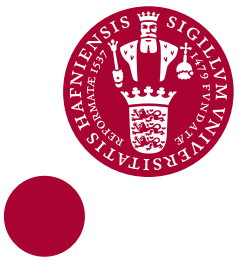
UNIVERSITY OF COPENHAGEN



Curve Evolution in Subspaces and Exploring the Metameric Class of Histogram of Gradient Orientation based Features using Nonlinear Projection Methods

Tatu, Aditya Jayant

Publication date:
2010

Document version
Early version, also known as pre-print

Citation for published version (APA):
Tatu, A. J. (2010). *Curve Evolution in Subspaces and Exploring the Metameric Class of Histogram of Gradient Orientation based Features using Nonlinear Projection Methods*. Faculty of Science, University of Copenhagen.

Download date: 05. May. 2023

Curve Evolution in Subspaces
and
Exploring the Metameric Class of
Histogram of Gradient Orientation based Features
using
Nonlinear Projection Methods

Aditya Tatu

The Image group, Department of Computer Science,
Faculty of Science, University of Copenhagen

January, 2010

Acknowledgement

Four years have gone onto getting this dissertation ready. My supervisors Prof. Mads Nielsen and Asst. Prof. François Lauze have played a major role in its development. A big thank you to them. With his considerable experience and knowledge about the field, Mads helped me in choosing appropriate reserch directions and gave general guidance. On the other hand, François helped me digest some tough math with his wonderful impromptu lectures and dealt with nitty-gritty details which engineers like me usually ignore. I would also like to thank my former supervisor Ole Fogh Olsen for accepting me as a Ph.d student and bearing with me for a year. I am thankful to Prof. Benjamin Kimia who took time out of his extremely busy schedule at the LEMS lab, Brown University, USA to guide me during my stay abroad in the winter of 2008.

The Image groups at ITU and DIKU have been entertaining and interesting groups to work with. I am surely going to miss the fun in the 'evil Ph.d office'. Several of my colleagues: Stefan, Aasa, Chen, David, Sune, Kim, to name a few, have had a positive impact on my work directly or indirectly. Camilla, Dina, Elizabeth and Gitte have made the University administration affairs as simple as they could be.

On the personal front, I've been fortunate to make some wonderful new friends in this period: Saurabh, Anil, Rajeev, Navin, Chandrashekar, Vishal & Kinjal, Prasad and family, Sriram & Sirisha, Srinivas, Kumar & Keshavpriya, Chaitanya Chandra, Gopal & Jhinuk, Matu uncle & aunty, Pamela, Vasanth & Ramya and many more. The Krishna temple in Vanlose has been a fabulous support system. I am grateful to everyone associated with the temple for many unforgettable moments.

Ratnik and Pratik, my friends in India, have always been 'online' to help me whenever needed.

Finally, I would like to thank my wonderful family: Jayant & Vaidehi (my parents), Rohit & Alpa (my brother and sister-in-law) and the little star of the family, Tanay (my nephew) for always encouraging and supporting me, and much more.

Aditya Tatu

Abstract

This thesis deals with two unrelated issues, restricting curve evolution to subspaces and computing image patches in the equivalence class of Histogram of Gradient orientation based features using nonlinear projection methods.

Curve evolution is a well known method used in various applications like tracking interfaces, active contour based segmentation methods and others. It can also be used to study shape spaces, as deforming a shape can be thought of as evolving its boundary curve. During curve evolution a curve traces out a path in the infinite dimensional space of curves. Due to application specific requirements like shape priors or a given data model, and due to limitations of the computer, the computed curve evolution forms a path in some finite dimensional subspace of the space of curves. We give methods to restrict the curve evolution to a finite dimensional linear or implicitly defined nonlinear subspace of curves. We also deal with cases where a non-Euclidean metric is induced on such a subspace. We build differential geometric tools like the Exponential map and Log map which are essential for the study of such nonlinear spaces. We demonstrate these tools on a particular implicitly defined subspace, the N-links bicycle chain space, i.e. the space of curves with equidistant neighboring landmark points. This in itself is a useful shape space for medical image analysis applications.

The Histogram of Gradient orientation based features are many in number and are widely used in applications like object recognition which is a vital component of any computer vision system. In order to get some intuition behind their success, we attempt to explore the metameric class of a basic version of such features. We characterize such an equivalence class using implicitly defined constraints over the statistical moments of the gradient orientations. This is another case for use of nonlinear projection methods since such an equivalence class is nonlinear. We use an approximation of the Exponential map developed in the first part of the thesis to evolve a given patch in the equivalence class. Specifically, given two initial visually different patches, we evolve one patch into another patch that visually looks like the other given patch, while still preserving its Histogram of Gradient orientation features.

Contents

Acknowledgement **i**

Abstract **iii**

1 Introduction **1**
1.1 Curve evolution . . . 1
1.2 Shape spaces . . . 3
1.3 Curve evolution in subspaces . . . 8
1.4 Histogram of Gradient orientation based features . . . 10
1.5 Organization of the thesis . . . 11

2 Curve Evolution in Subspaces **13**
2.1 Introduction . . . 13
2.2 Evolution in Subspaces . . . 15
2.3 B-Spline subspace . . . 16
2.4 Experiments . . . 19
2.5 Conclusion . . . 19

3 Bicycle chain shape manifolds **23**
3.1 Introduction . . . 23
3.2 Preshape manifolds . . . 26
3.2.1 n-Links Bicycle Chain Manifolds . . . 26
3.2.2 Removing Translation and Scaling . . . 26
3.2.3 Geodesic Generalized Procrustes Analysis . . . 27
3.2.4 Principle Geodesic Analysis (PGA) . . . 27
3.3 Geodesics on the manifold; the Exp- and Log-map . . . 28
3.3.1 Shooting method . . . 29
3.3.2 Path straightening . . . 30
3.4 Experiments . . . 31
3.4.1 Illustrative example . . . 31
3.4.2 Vertebra shapes . . . 32

3.5 Conclusion 33

4 Curve Evolution in Implicitly defined Curved Subspaces 37
4.1 Introduction 37
4.2 Background & Related work 38
4.3 Evolving curves by solving an Ordinary differential equation in a subspace 40
4.4 Space of Curves 41
4.5 A series of Projection steps 42
4.5.1 The projection P_C^V 42
4.5.2 The map P_C^M 43
4.5.3 The map P_C^g 44
4.6 Working with Coordinates 45
4.7 An Example: Euclidean Shortening Flow for Splines with $\mathbb{R}^2$-Equidistant Node Points . 47
4.8 An Application: Restricted Geodesic Active Contours 48
4.9 Experiments 49
4.10 Conclusion 50

5 On Restricting Curve Evolution to Finite Dimensional Implicit Subspaces with Non-Euclidean Metric 61
5.1 Introduction 61
5.2 Projected Curve Evolution 64
5.2.1 Projecting to the tangent space of S^N 64
5.2.2 Projection to tangent space of V 65
5.2.3 Exponential map on the submanifold V 65
5.3 Log map using Parallel transport 67
5.3.1 Stereographic projection 70
5.4 Subspace of Equidistant neighboring node point spline curves 72
5.5 Experiments 79
5.6 Conclusion and Future work 79

6 Metameric class of Histogram of Gradient Orientation features 85
6.1 Introduction 85
6.2 Equivalence class of HOGO features 86
6.3 Constrained optimization 87
6.3.1 Numerical algorithm 90
6.4 Experiments 90
6.5 Conclusion 94

7 Discussion and Conclusion 99

A Definitions from Differential geometry 101

References **105**

Chapter 1

Introduction

Projection based methods are quite commonly used in the areas of image analysis and computer vision. In this thesis, we build projection schemes over nonlinear spaces to deal specifically with two problems: restricting curve evolution to given subspaces and measuring information content encoded in Histogram of Gradient orientation based features.

Before describing the problems in depth and the method used to alleviate them, we begin with a gentle introduction to curve evolution and shape theory, followed by another introduction to Histogram of Gradient orientation based features.

1.1 Curve evolution

Curve evolution arises in various applications like tracking interfaces between mediums, computing geodesics between shapes, segmenting objects using active contour algorithms. A parametric curve (in $\mathbb{R}^2$) is evolved by propagating all points on the curve according to a given velocity vector function $V(p,t)$, where p is the parameter of the curve and t is the evolution time. The family of curves $\mathcal{C}(p,t)$ satisfies the following partial differential equation:

$$\frac{\partial}{\partial t}\mathcal{C}(p,t) = \overrightarrow{V}(p,t). \tag{1.1}$$

The velocity vector at a point on the curve can be decomposed into a tangential and a normal component to the curve:

$$\overrightarrow{V} = V_N\overrightarrow{N} + V_T\overrightarrow{T},$$

where V_N is the scalar normal velocity component, $\overrightarrow{N}$ is the unit normal to the curve, V_T is the scalar tangential velocity component and $\overrightarrow{T}$ is the unit tangential vector to the curve. The tangential component of the velocity affects only the parameterization of the curve and the not its image [23], which reduces the curve evolution equation to

$$\frac{\partial}{\partial t}\mathcal{C}(p,t) = V(p,t)\overrightarrow{N}(p,t). \tag{1.2}$$

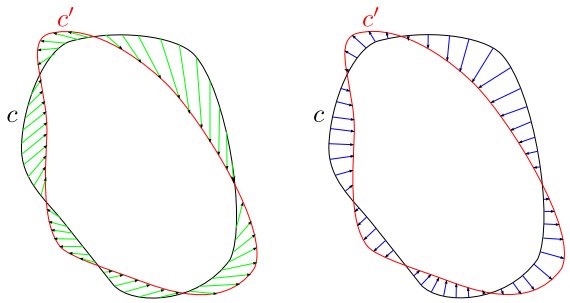


Figure 1.1: Curve evolution. (left) Velocity with tangential and normal component. (right) Velocity with the normal component also gives the same curve, with different parameterization.

where V is the scalar speed in the normal direction. As shown in Figure1.1, on the left is shown a curve evolving according to a velocity with both tangential and normal component, while on the right the same curves are obtained with the corresponding normal velocity vector field.

The velocity is defined according to the application at hand. Sometimes, however, the velocity is not 'designed', but arises as the gradient of an energy functional to be minimized. Let $\mathcal{E}(\mathcal{C})$ be an energy functional to be minimized over the space of curves. Then

$$\frac{\partial}{\partial t}\mathcal{C}(p,t) = -\frac{\partial}{\partial \mathcal{C}}\mathcal{E}(\mathcal{C}) = -V(p,t)\overrightarrow{N}(p,t)$$

is a gradient descent scheme to search for a minima.

A particular example of a curve evolution scheme that stands out is the curvature flow, given as

$$\frac{\partial}{\partial t}\mathcal{C}(p,t) = \kappa(p,t)\overrightarrow{N}(p,t), \tag{1.3}$$

where κ is the curvature of the curve and $\overrightarrow{N}$ is the inner unit normal vector to the curve. This flow has interesting properties like

- A simple curve, under the curvature flow, evolves into a convex curve and vanishes at a round point [31].
- Under the curvature flow, the curve does not self intersect [31].
- The curvature flow is a curve shortening flow, i.e., it is a gradient descent on the length of the curve.
- Two curves under curvature flow, one completely contained in the other, will not intersect during their evolution.

- The curvature flow simplifies curves, i.e., the variation of curvature of a curve reduces with time [72].

Topological changes in the curve cause problems while evolving parametric curves. One way to handle topological changes is to evolve the corresponding level set function of the curve [72]. In this work, however, we exclusively work with parametric curve evolution.

Boundaries of objects in images are usually piecewise continuous curves. These curves can be represented by different ways like: parameterized curves, landmark points, distance functions, fourier descriptors, chain codes etc. We explain some of them in the next paragraph. We would like to know the geometry of the object boundary, irrespective of the representation, since the geometry gives us vital information regarding the shape of the object. Typical problems in shape theory include extracting shape information from the representation, methods to compare shapes, generating new shapes, computing shape statistics. We now give a brief survey of work in shape modeling and shape metrics relevant to our study.

1.2 Shape spaces

The shape of an object or configuration of points is defined by Kendall [42] as whatever information is left after removing translation, rotation and scaling. The first study of shapes was done by the biologist and mathematician D'Arcy Thompson in [81], wherein he studied the shape of a collection of similar looking organisms, by studying the transformation required to map one organism to another. The modern theory of shape is credited to Kendall [42] and Bookstein [5]. Some applications of shape theory are: shape comparison [5, 32], shape recognition [74] and shape statistics and classification [24, 60] etc.

We first survey some of the prevalent ways of representing shape data and describe the methods to extract shape information from them.

- Kendall's shape space:

  We restrict the discussion of Kendall's shape space to a collection of landmarks in $\mathbb{R}^2$. An object is represented by k landmark points in $\mathbb{R}^2$, say $(x_1, x_2, \ldots, x_k)$. In order to obtain information about the shape, we need to remove the effect of translation, rotation and scaling from the data. Translation is filtered out by moving the centroid $\bar{x} = \frac{1}{k}\sum_{i=1}^{k} x_i$ to the origin. This reduces the dimensions of our space of configurations by two. Effects of scaling is eliminated by scaling all the points in a given configuration so that the size defined as $S = \sqrt{\sum_{i=1}^{k}(x_i - \bar{x})^2}$ becomes one. Thus the configuration is now a point on the unit sphere $\mathbb{S}^{2(k-1)-1}$. This sphere is called the *preshape space*. The *shape space* is the quotient space of the preshape space with the special orthogonal group $SO(2)$ (the group of rotations of the configuration in $\mathbb{R}^2$). Thus a shape is an equivalence class on the preshape space. The distance between two given shapes is the minimum distance between the corresponding equivalence classes. Given two preshapes S_1, S_2, the geodesic

distance between the shapes $[S_1], [S_2]$ is given as,

$$\begin{aligned} d([S_1],[S_2]) &= \inf_{R,R'} d(RS_1, R'S_2), \quad R, R' \in SO(2) \\ &= \inf_{R} d(RS_1, S_2), \quad R \in SO(2) \\ &= \inf_{R} \{\cos^{-1}(\langle RS_1, S_2 \rangle)\} \end{aligned}$$

We have $0 \leq d \leq \pi$. Kendall also discusses shape densities and shape measures for landmark configurations in a general m dimensional space, but this is beyond the scope of this thesis.

- Space of smooth parametric curves:

  In [51], the authors study the space of smooth closed planar curves. Strictly speaking this is not a shape space as defined by Kendall [42], nonetheless, a lot of concepts explained are applicable to our work on curve evolution.

  The objects of their interest are

$$B_e(\mathbb{S}^1, \mathbb{R}^2) = \text{Emb}(\mathbb{S}^1, \mathbb{R}^2)/\text{Diff}(\mathbb{S}^1)$$

$$B_i(\mathbb{S}^1, \mathbb{R}^2) = \text{Imm}(\mathbb{S}^1, \mathbb{R}^2)/\text{Diff}(\mathbb{S}^1)$$

  where $\text{Emb}(\mathbb{S}^1, \mathbb{R}^2), \text{Imm}(\mathbb{S}^1, \mathbb{R}^2)$ are the space of embeddings and immersions from $\mathbb{S}^1$ to $\mathbb{R}^2$ respectively, and $\text{Diff}(\mathbb{S}^1)$ is the group of diffeomorphisms of $\mathbb{S}^1$ to itself, i.e. the group of reparameterizations. This identifies curves that are reparameterizations of each other. They prove that the standard L_2 reparameterization invariant inner product between normal curve deformations $h(s), k(s)$ of a curve $c(s)$, given by

$$\langle h, k \rangle_{H^0} := \int_0^L h(s)k(s)|c'(s)|ds \tag{1.4}$$

  degenerates, i.e., the geodesic distance between any two curves in B_e or B_i is zero. Intuitively, one can reduce the cost of deformation of a curve into any other curve using higher frequency deformations, which in a limit can be made to go to zero. Mumford and Michor [51] propose to use a curvature dependant inner product

$$G_c^A(h, k) := \int_0^L (1 + A\kappa(s)^2)\langle h(s), k(s) \rangle |c'(s)|ds \tag{1.5}$$

  where $\kappa(s)$ is the curvature of the curve at $c(s)$. In a more recent paper [52], they use the Weil-Petersen metric. Both these inner products limit the amount of high frequency deformations.

  In [78], the authors propose to use the Sobolev inner product on normal curve deformations:

$$\langle h, k \rangle_{H^n} := \langle h, k \rangle_{H^0} + \lambda \langle h^n, k^n \rangle_{H^0} \tag{1.6}$$

where h^n denotes the n^{th} order derivative with respect to the curve parameter and λ is a strictly positive constant. Due to the presence of derivatives of the deformation field in the inner product, one cannot introduce very high frequency deformations, thereby preventing the metric from degenerating.

In a more group theoretic approach, Younes [87] uses the metric of a group of infinitesimal curve deformation vector fields to induce a metric on the space of smooth planar curves. They assume that any curve can be deformed to any other curve using elements from the group of deformations. The geodesic distance between two given curves is the minimum distance between the elements from the group used to deform one curve to another. The definition of distance also depends on the spatial derivatives of the deformation field, thereby restricting the frequency of deformation allowed.

- Finite Fourier descriptors:

  Fourier descriptors have been used for representing plane closed curves since Zahn and Roskies [88]. Some applications of Fourier descriptors include character recognition [61, 30], shape coding [14], shape classification [39] and object recognition [67]. We focus on the shape space described in Srivastava *et al.* [77]. A closed parametric curve can be represented by a complex function $c(s) = x(s) + jy(s)$. Since $c(s)$ is periodic, we get a Fourier series representation of $c(s)$,

$$c(s) = \sum_{n=-\infty}^{\infty} a_n \exp\left(j\frac{2\pi ns}{L}\right)$$

  where L is the period. The Fourier coefficients are given as,

$$a_n = \frac{1}{L}\int_0^L c(s)\exp\left(-j\frac{2\pi ns}{L}\right)\, ds$$

  Instead of using Fourier coefficients of the co-ordinate functions, Srivasatava *et al.* [77] propose to use the Fourier descriptors of the direction function $\theta(s)$ (i.e. $\theta(s)$ is the angle between the tangent to the curve at point $c(s)$ and the positive x axis) and the Fourier descriptors of the curvature function $\kappa(s)$. This representation is invariant to translation. In the former case, to quotient out the group of rotations in the plane, they constrain the direction function $\theta(s)$ with the condition:

$$\frac{1}{2\pi}\int_0^L \theta(s)ds = \pi$$

  Any L_2 function that is a direction function of a closed curve also satisfies the constraint:

$$\int_0^L \exp(j\theta(s))ds = 0$$

  The functions $\theta \in L_2$ satisfying these constraints form the preshape space. The shape space is defined as the quotient of the preshape space with the group of reparameterization $\mathbb{S}^1$. In this

space geodesics are computed using a combination of a gradient descent approximation to the Exponential map and a shooting method to compute the Log map.

Instead of using the shooting method to compute the Log map and the geodesics, Schmidt *et al.* [68] use an initial path between the shapes and carry out a length minimizing gradient descent step to compute the geodesic. However computing an initial path between two given curve may be difficult in some cases.

- Bicycle chain shape models [76]

  In medical image analysis, shapes are often represented as a collection of landmarks and pseudo-landmarks. Pseudo landmarks are usually required to be equidistant to their neighbors. In [76], we propose a shape model that requires that all points on the curve be equidistant to its neighbors. Mathematically, this condition for an open curve is written as a function $F : \mathbb{R}^{2n} \to \mathbb{R}^{n-2}$ given as

  $$F_i(P_1, ..., P_n) = d_{i+2,i+1} - d_{i+1,i}, \;\; i = 1, .., n-2 \tag{1.7}$$

  where $d_{i,j} = (x_i - x_j)^2 + (y_i - y_j)^2$ is the squared euclidean distance between points P_i and P_j. This gives us an implicitly defined space $V = F^{-1}(0) \subset \mathbb{R}^{2n}$. Since we do not have an explicit parameterization of V, some basic differential geometric tools like the Exponential map are not trivial to compute. These are essential for many applications like shape matching and segmentation of shapes living in this particular nonlinear shape space V.

  In this thesis, we develop tools like the Exponential map and Log map, and show how to do curve evolution and nonlinear statistics on this implicitly defined finite dimensional shape space. Since the members of this space are 1-dimensional submanifolds of $\mathbb{R}^2$ (curves) and not just points in $\mathbb{R}^{2n}$, we also develop some of the tools when our space is equipped with a non-Euclidean metric, which can be induced from the space of immersions or embeddings. In particular we use the standard H^0 inner product (1.4) defined on the spaces B_e or B_i. Since the space is finite dimensional, the metric does not degenerate as in [51].

- Diffeomorphic deformation curve mapping:

  Two planar curves can be related to each other via a diffeomorphism. Glaunés *et al.* [27], search for an optimal diffeomorphism that maps one shape to another. Diffeomorphisms are defined on the interior of an open domain $\Omega \subset \mathbb{R}^2$ in which the shapes lie, with the boundary condition that the diffeomorphism is identity on and beyond the boundary $\partial\Omega$. Jordan curves are considered and the diffeomorphisms model deformation of one shape (the curve and its interior) into another. Large deformations ϕ are obtained by composing several diffeomorphisms and each such diffeomorphism is given by a flow equation

  $$\frac{\partial \phi_t}{\partial t} = u_t \circ \phi_t \quad \phi_0 = Id$$

  where u_t belongs to a Hilbert space of vector fields at every time instant $t \in [0, 1]$. The optimal diffeomorphism is the solution of the flow equation at $t = 1$, which minimizes a cost function

containing a regularization term and a data matching term. The regularization term is defined as,

$$R(\phi) = d_G(Id, \phi)^2$$

where d_G is the geodesic distance on the infinite dimensional manifold of diffeomorphisms. The cost function to be minimized in written in terms of the vector field u which makes it computationally manageable as it belongs to a vector space. Other examples of work based on this strategy are [26, 54].

- Shape representation using Conformal maps.

  The Riemannian mapping theorem states that there exists a conformal mapping ϕ_-, unique upto a Möbius transformation, between the unit disk Δ_- and the interior of any simple planar closed curve Γ (denote its interior by Γ_-). Using the inverse transform $T(z) \rightarrow \frac{1}{z}$, one can also define a similar conformal mapping ϕ_+ between the exteriors Δ_+ and Γ_+. These mappings restricted to the unit circle Δ, gives a diffeomorphism $\Psi = \phi_+^{-1} \circ \phi_-$ on $\mathbb{S}^1$ to itself which is called the *fingerprint* of the curve Γ. Two preshapes map to the same diffeomorphism only one shape is related to the other by a translation and scaling transform. Sharon and Mumford [73] give computational methods for computing the fingerprint from a given curve and vice versa. The quotient of Diff$(\mathbb{S}^1)$ with the Möbius transformations form the equivalence class of shapes. This quotient is then equipped the Weil-Petersen metric to compute geodesics between two given shapes.

- Distance functions:

  Distance functions were used in image processing as a way of representing a curve [58]. Charpiat *et al.* [11] present a way to compare shapes by representing the shapes as distance functions. A Hausdorff metric is defined on the shapes which can be computed using the distance functions. The set of shapes with bounded curvature is considered. In order to deform one shape into another, an energy functional (based on Hausdorff and other distances) is defined on the shape space and a gradient descent is done from one shape to another. Since Hausdorff distance is not differentiable, a smooth approximation is used instead.

  In a different but related approach by Gorelick *et al.* [29], a shape S (a simple closed planar curves) is represented as a function in its interior which gives the mean time for a random walk to reach the boundary from that point in the interior. The function can be computed as a solution to the Poisson equation:

$$\begin{aligned} \Delta U(x,y) &= -1, \quad (x,y) \in S \\ U(x,y) &= 0 \quad (x,y) \in \partial S \end{aligned}$$

  Using this representation one can divide the shape into convex and concave subparts, compute its skeleton (different from medial axis) and deduce various other properties of the shape. The authors use these properties to classify shapes.

- Shock Graphs:

  Shock graph representation of shapes [74] is one of the few methods that does not directly represent the object boundary. It is based on Blum's medial axis [4] which is the locus of shock points when solving the grassfire curve evolution equation for the given shape. In addition to the location of the singularities, shock graphs also store the direction of flow of singularity and the instantaneous velocity at each shock point. Deformation of a shape could either change some of the properties like location and curvature of individual links of the shock graph, or change the topology of the shock graph. The topology changing transformations are enumerated by Giblin and Kimia [25] and hence the dimensionality of the space in which one has to search for paths of deformation from one graph topology to other is finite. The cost of deformation is defined such that it discourages deformations that add a feature and remove another and is called the Edit-distance [70]. In other words, geodesics are simplifying deformation paths.

1.3 Curve evolution in subspaces

Curve evolution has become a standard tool in Computer Vision. It has been used for tracking interfaces, registration of shapes, active contour algorithms for segmentation, etc. [38, 72, 44, 3]. Some of the core problems in shape theory are in one way or the other related to infinitesimal shape deformations. Curve evolution is a natural choice to study shape deformation whenever the shapes are given as curves. Curves can be manipulated implicitly as the zero-level set of a given function for instance, or explicitly via a parametrization. Here, we focus on the latter.

As discussed earlier, curve evolution arises out of a gradient descent equation to minimize some energy functional. For example, consider the following energy functionals:

1. Length of the curve:

   $$\mathcal{L}(C) = \int_0^p |C'(s)| ds$$

   The gradient descent equation gives the classical Curvature flow [31]:

   $$\frac{\partial C}{\partial t} = -\frac{\partial \mathcal{L}}{\partial C} = \kappa N$$

   where κ is the curvature of the curve.

2. Geodesic active contour energy:
   Geodesic active contours are used for segmentation of object boundaries [9]. The energy of a curve is defined as:

   $$\mathcal{E}(C) = \int_0^p g(|\nabla I(C(q))|)|C'(q)|\, dq$$

   where $I(x)$ is the image and $g : [0, +\infty[\rightarrow \mathbb{R}_+$ is a strictly decreasing function with $g(r) \rightarrow 0$ as

$r \to +\infty$. The gradient descent equation is given by

$$\frac{\partial C}{\partial t} = -\frac{\partial \mathcal{E}}{\partial C} = g(I)\kappa N - (\nabla g \cdot N)N$$

3. Sum of Squared distances between curves:

   The intrinsic mean curve [60, 24] of a collection of curves is given by the minimizer of the sum-of-squared distance function:

$$\mathcal{D}(C) = \frac{1}{2N}\sum_{i=1}^{N} d(C, C_i)$$

   where $C_i, i = 1, 2..., N$ are the given curves whose mean has to be computed. The distance $d(C, C_i)$ is the geodesic distance between curves C and C_i on the manifold of curves considered. The gradient descent equation [37] is given by:

$$\frac{\partial C}{\partial t} = -\frac{\partial \mathcal{D}}{\partial C} = \frac{1}{N}\sum_{i=1}^{N} Log_C(C_i)$$

   where $Log_C(C_i)$ is the logarithm map on the given manifold of curves at the curve C of the curve C_i.

In general, the family of curves $C(., t), t \in [0, \infty)$ obtained from curve evolution PDE's form a path of curves in some infinite dimensional space of curves.

Technical considerations may require us to restrict the manifold of curves in which the evolution takes place, for instance by imposing specific constraints on the curve representations or specifying shape priors for segmentation. When curve evolution is simulated on a computer, due to the inevitable discretization involved, the evolution may not follow the theoretical flow intended by the equation, and is carried out in some finite dimensional subspace of the generic space of curves. The subspace in which the curve evolution is required to be restricted could be a linear or nonlinear finite dimensional subspace of the space of all curves. At each iteration of the curve evolution equation, we need to evolve the curve in the direction given by the velocity vector, which in this case is given by the gradient of the energy functional. This vector may take the curve out of the desired subspace, hence projection methods are required to keep the evolution within the subspace. Restricting to nonlinear subspaces requires more complicated projection methods. Projection to the tangent space is not enough, as a discrete step in the tangential direction will evolve the curve out of the desired subspace.

In this thesis, we discuss how to build shape/curve processing tools like curve evolution and computing mean shape on a given linear or nonlinear finite dimensional space of curves. In particular, we are interested in the finite linear subspace of spline curves and the Bicycle chain shape space discussed in the previous section. The latter is an example of a nonlinear implicitly defined subspace, which is more complicated then a parameterized subspace.

In order to compute distance between given shapes, one needs to define an inner product on the

shape space. The definition of inner product is crucial to shape processing [12]. For example, computing the intrinsic mean of a collection of members from B_e(space of embedded curves) using the H^0 inner product does not make sense, since it is degenerate. We build tools essential for curve evolution when the implicitly defined nonlinear subspace is equipped with a non-Euclidean inner product. Specifically, we embed the Bicycle chain shape space in the space of embeddings and use the induced H^0 metric on it to evolve curves.

We give two different methods to restrict curve evolutions to the implicitly defined nonlinear subspace. The first method to restrict curve evolution involves approximating the Exponential map, using a gradient descent approach, much like in [77]. The method consists of moving on the tangent space and then doing a gradient descent to reach the nonlinear subspace. We develop this approximation also in the case of a non-Euclidean inner product on the subspace, using the fact that the definition of gradient depends on the inner product.

The other method is based on tools from differential geometry. We use basic differential geometric maps like the Exponential map and Log map for the purpose of curve evolution. One can compute the geodesics and the geodesic distance between two shapes using these two maps. We develop the tools in the case where the induced metric on the subspace is non-Euclidean and demonstrate them in applications of curve evolution.

We use the approximation of the Exponential map to explore the equivalence class of Histogram of Gradient orientation based features, which we now introduce.

1.4 Histogram of Gradient orientation based features

A feature is an information extracted from an image and is specific to the application intended. It is usually the result of a neighborhood operation, for example, blobs, edges or corners. One major area where features are important is object recognition. It is a vital component in any computer vision system. A number of features have been developed to suit various scenarios and problems in object recognition. If the object has to be identified in different images, the set of features should be invariant to some of the transformation that are expected in the system, for example, illumination changes, viewpoint transformations etc. Some of the widely used features are based on the concept of Histogram of Gradient Orientation (henceforth denoted as HOGO).

Given an image I defined over $\Omega \subset \mathbb{R}^2$, the HOGO feature at a point $p \in \Omega$ is computed by first calculating the image gradient in its neighborhood D of size $m \times m$ pixels. The 8-bin angular histogram of the gradient orientations in D gives the HOGO feature at the point p.

Many versions of this feature have come up with different modifications. Edge orientation histograms [79], shape contexts [2], SIFT [48], PCA-SIFT [86], GLOH [53], Histogram of Oriented Gradients [17] are few of them. A comparative study of the performance of some of the above mentioned features and others is given in [53].

In SIFT, histograms are computed at extremas of scale and space laplacian of the image on 16 blocks of 4×4 pixels in a 16×16 neighborhood. Dalal and Triggs [17] compute the histograms at overlapping

neighborhoods of 10×10 pixels all over the image. One can also use the magnitude of gradient as weights in the histogram.

In Chapter6, we focus on the very basic version we just described. We try to find the size of the equivalence class of patches that have the same HOGO feature. Lillholm *et al.* [50] compute image patches from the metameric class of linear gaussian scale space features. We need to represent the histogram of directional data in order to charaterize the equivalence class. Given all statistical moments of a random variable, one can compute its distribution function [59]. We use a finite number of statistical moments to approximate the histogram. The p^{th} statistical moment of some directional data ϕ defined on Ω is a complex number

$$r_p = \frac{1}{|\Omega|}\left[\int_\Omega \cos(p\phi)dx\,dy + i\int_\Omega \sin(p\phi)dx\,dy\right]$$

Constraining a finite number of these moments to particular values gives us an implicit representation of the equivalence class of a given patch. It is easy to see that such a space is infinite dimensional. To explore or move around in this equivalence space, one needs an effective nonlinear projection technique. The Exponential map would be ideal, but computing an infinite dimensional version of the one we compute in Chapter5, seems very complicated, hence we use its approximation from Chapter4 using gradient descent.

1.5 Organization of the thesis

We begin with restricting curve evolution to linear finite dimensional subspace of the space of curves in Chapter2. We focus particularly on the space of closed spline curves with N node points. We also propose a new time-step for this particular evolution. In Chapter3, we work on open curves in Bicycle chain shape space and compute Principle Geodesic Analysis subspaces for representing vertebra shapes on it. We continue the work on Bicycle chain shape space in Chapter4, this time with closed planar curves, and in the case when the induced metric on the subspace is non-Euclidean. In order to project curve evolution into the nonlinear subspace, we use an approximation of the Exponential map using gradient descent to project from the tangent space to the subspace. In Chapter5, we develop the Exponential and Log maps using optimal control theory on Bicycle chain shape space. We use these tools to restrict curve evolution and to compute geodesics between two given shapes. We give basic definitions from Differential geometry in AppendixA. We attempt to get some intuition behind the success of Histogram of Gradient orientation based features in Chapter6. To explore the metameric class of patches with the same Histogram of Gradient Orientation features, which is an infinite dimensional implicitly defined subspace, we use the approximation of the Exponential map using gradient descent. We conclude with some discussion and future research directions in Chapter7.

Chapter 2

Curve Evolution in Subspaces

This chapter has been published as *Curve Evolution in Subspaces* in the proceedings of *First International Conference on Scale Space and Variational Methods in Computer Vision, 2007*.

2.1 Introduction

Curve evolution methods have been used for several applications like propagating interfaces, object segmentation, image inpainting etc. [44, 72]. They often come out of minimization problems in which an objective function cannot be directly maximized/minimized, and we need an iterative scheme to reach the maximum/minimum of the cost function. The basic form of curve evolution is given as

$$\frac{\partial \mathbf{C}}{\partial t}(p,t) = V\mathbf{N} \tag{2.1}$$

where V is the velocity vector giving the amount of deformation of the curve, $\mathbf{N}$ is the inner unit normal to the closed curve $\mathbf{C}$, p is the parameter of the curve and t is the time of evolution [44]. V is chosen depending on the task to be performed. Depending on the flow chosen, the initial curve changes to other intermediate curves in some specific space of all curves, generally L^2 or Sobolev space*.

We focus on the evolution of curves restricted to a given subspace of all curves. This is important because some application may require all intermediate curves to lie in some given subspace specified by some set of shapes, for example the space of all curves of a human right hand boundary and its simplifications [32]. In any case, while implementing Equation 2.1 on a computer, one has to discretize the PDE, and hence the evolution is implicitly carried out in some subspace of the specified space of curves considered, for example L^2, and not in the entire space. One often used approach is to approximate the curve evolution and neglect the fact that the intermediate curves obtained in the discrete implementation may be different then the actual intended curve evolution. The subspace, obviously depends on the type of discretization used.

*For example, if the curve evolution is dependant on the curvature, then we would like the space of curves to be a Sobolev space, say $W^{2,2}$

Figure 2.1.(left), depicts curvature flow of an ellipse, where curvature is approximated by central difference derivatives, and the curve is considered as a polygon with given vertices. On its right, curvature flow of the same set of points, considering a 3^{rd} order B-spline interpolation, with our approach is shown. Note that, in both cases, we evolve just the vertex or control points marked as '*'. Clearly, it advocates study of curve evolution in subspaces.

In this paper, we formulate the curve evolution equations, given the fact that we need to restrict the curves to some subspace of all curves. Here, we work with finite dimensional linear subspaces with a Hilbertian structure. The subspace may be specified by giving some example shapes from the subspace, or by specifying the basis vectors of the subspace. We do not impose the restriction of orthogonality on the basis vectors.

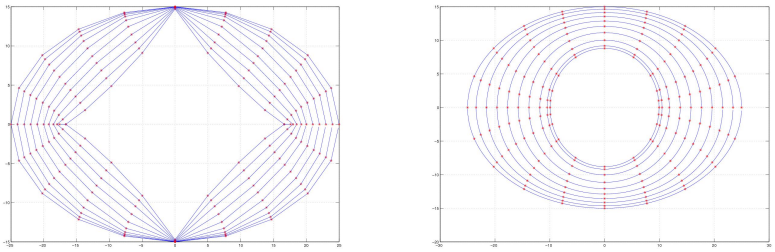


Figure 2.1: (left) Curvature flow of points on an ellipse. The ellipse is approximated as a polygon. The properties of the curve like the curvature are computed using the central difference derivatives. (right) Curvature flow of the same points on the ellipse. The curve is approximated by 3^{rd} order B-splines and evolution is carried out using our approach. We can see that the flow shown in the left, is erroneous, and the one to the right is as expected, evolving towards a round point

Similar attempts to restrict curve evolution to some particular subspace are given in [66], where Sapiro and Tannenbaum restrict the curvature flow to curves with a given constant area, Bruckstein *et al.* [8] give methods to approximate properties of curves represented as planar polygons in order to carry out curve evolution. There is related work in the area of statistical shape priors in active contours. Cremers *et al.* [16] restrict the curve in the linear shape space formed by some example shapes. In [85], the active contours are constrained to a subspace having desired regional properties, similar to the object that is to be segmented.

This paper is organized as follows. In section 2.2, we describe the problem in general. In section 2.3, we take up an example of a particular subspace, namely the B-spline subspace, and give details of our method. We give experimental results in section 2.4 and conclude in section 2.5.

2.2 Evolution in Subspaces

Let Δ be the desired subspace of the space of all curves considered, say Γ, i.e. $\Delta \subset \Gamma$. Given a curve $C^i \in \Delta$, and given a curve, $C^{i+1} = C^i + V^i\mathbf{N}^i dt$, we need to find a curve $\hat{C}^{i+1} \in \Delta$ which best approximates C^{i+1}. The velocity vector V may be such that, it may move the curve out of the desired subspace. Hence it is clear that, it is not possible to exactly follow the given curve evolution in a subspace, unless the velocity vector forces the curve to stay in the given subspace. So the best alternative is to follow the flow as accurately as possible, i.e., to project a deformed curve back into the desired subspace. The Classical Projection theorem [49] states that the best approximation to a given vector is its orthogonal projection into the desired subspace. We follow this principle in our formulation. Given a curve evolution scheme as in Equation 2.1, our approach is to carry out the corresponding projected evolution given as

$$\frac{\partial \mathbf{C}}{\partial t} = proj(V\mathbf{N})$$

Using Forward difference scheme:

$$\mathbf{C}^{i+1} = \mathbf{C}^i + proj(V^i\mathbf{N}^i)dt$$

where $proj(.)$ is the projection operator on the given subspace Δ. The projection operator could depend on the curve $\mathbf{C}^i$ if the subspace is non-linear. Here, we consider only linear subspaces (vector spaces). We allow our curve to deform and move out of the subspace at every iteration, then we project it back into the desired subspace. Let $\{B_k\}_{k=0,1..N-1}$ be the set of basis vectors of the subspace. A curve C in the subspace is represented as $\mathbf{C} = \sum_{k=0}^{N-1} a_k\ B_k$. In order to evolve this curve according to Equation 2.1 in the subspace, we use

$$\mathbf{a}^{i+1} = \mathbf{a}^i + G^{-1}.P_B(V^i\mathbf{N}^i)^t dt \tag{2.2}$$

where $P_B(V^i\mathbf{N}^i) = [\langle V^i\mathbf{N}^i, B_0\rangle\ \langle V^i\mathbf{N}^i, B_1\rangle\ ...\ \langle V^i\mathbf{N}^i, B_{N-1}\rangle]$, $\mathbf{a}^i = [a_0^i a_1^i ... a_{N-1}^i]$ are the parameters of the curve at iteration i, G^{-1} is the inverse Gram matrix [49] (also explained in section 2.3), required in case our basis vectors B_k are not orthogonal, $\langle ., . \rangle$ is the standard L^2 inner product and dt is a suitable time step (P_B^t indicates matrix transpose of P_B). The effect of choosing different inner products is discussed in details in Chapters 4 and 5 The deformation is projected onto the tangent space of the desired subspace of all curves. If the subspace is linear then the tangent space can be identified with the subspace. The new curve is given as, $C^{i+1} = \sum_{k=0}^{N-1} a_k^{i+1}\ B_k$. This problem is similar to constrained optimization problem over linear subspaces. It was shown by J. Rosen [65] that solving an optimization problem restricted to linear subspaces using Lagrange multipliers, is equivalent to solving the unconstrained problem and then projecting the solution orthogonally into the required subspace. The projection evolution will have a stationery point in case the evolution occurs in a direction orthogonal to the subspace. Two different curves trace out different evolution paths in the space of curves, when evolved according to the original equation, but may cross each other's trajectory when the evolution is carried out in some subspace, because of the projection. In case, we are given example shapes lying in a subspace, we compute the basis vectors by finding the eigenvectors of the covariance matrix, after

aligning the shapes.
Tuning the time-step dt is important. We consider the given curve evolution to be a gradient descent on a cost/energy function, say $Ł(\mathbf{C}(\mathbf{a}), t)$ where $\mathbf{C}$ is the curve, $\mathbf{a}$ are the coefficients of the basis vectors of the subspace Δ and t the time parameter. For example the curvature flow is a descent on the length of the curve. Then the curve evolution equation can be written as

$$\frac{\partial \mathbf{C}}{\partial t} = -\frac{\partial Ł(\mathbf{C}, t)}{\partial \mathbf{C}} = F(\mathbf{C}, t)$$

Using the Forward difference scheme:

$$\mathbf{C}^{i+1} = \mathbf{C}^i + F(\mathbf{C}, t)dt$$

Now if the cost function is purely quadratic, then the gradient is linear and one can fix an optimal value of the time step for the evolution. But for example in the curvature flow, the cost function is highly non-linear. The time-step has to be adapted to the shape of the cost function at each iteration. The time-step suggested in such cases is

$$dt = \frac{1}{max(||V\mathbf{N}||_{\mathbb{R}^2})}$$

When employed in our approach to curve evolution, it gives rise to oscillations in the curve at certain stage in the evolution. To overcome this, we use the information about the shape(curvature) of the cost function. Hessian of the cost function gives the information about the curvature of the cost function(surface). We would like to take small steps where the curvature of the cost function is high, and larger steps where it is low. The reason being, there are cost surfaces with different curvatures but similar gradient values near the optima. In such cases, if the curvature is not taken into account for the time-step the evolution may jump over the point of maxima/minima. So, one way to achieve this is to make the time step dependant on the inverse of hessian, $Hess(Ł(\mathbf{C}, t))$. But the cost surface may not have the same curvature in all directions. Taking a conservative approach, we use the maximum curvature at the current point in evolution, to decide the time-step. The eigenvalues of the hessian matrix approximate the principle curvatures of the cost surface [64]. Hence we use the time-step given by

$$dt = \frac{1}{|\lambda_{max}|} \tag{2.3}$$

In the next section, we derive the equations of curve evolution restricted to the subspace of closed spline curves with a given number of node points.

2.3 B-Spline subspace

We take an example of a particular subspace of interest and a particular curve evolution equation and derive our formulation for it. The following formulation applies equally well to other examples of linear subspaces and other curve evolution equations. We restrict the curvature flow to the space of curves

generated by n^{th} order B-spline basis functions.
The n^{th} order spline basis functions are given by

$$\beta^n(x) = [\beta^0(x) * \beta^0(x) * ...(n+1)times]$$

where $\beta^0(x)$ is the 0^{th} order spline function and $*$ denotes convolution. Note that the spline basis are not orthogonal. If n^{th} order spline is used, then the curve obtained belongs to C^{n-1} class of functions. This is important in cases where we need to carry out evolution of the curve with a velocity vector V that depends on some differential property of the curve. For example, to carry out the curvature flow of a curve, it becomes important that we should be able to compute the first and second derivatives of the curve with respect to its parameters, analytically. For details regarding B-spline functions, we refer the reader to [82, 83, 84]. We now derive the evolution equation for curvature flow of a curve, restricted to the B-spline subspace.
The curvature flow is given as

$$\frac{\partial \mathbf{C}}{\partial t}(p,t) = \kappa \mathbf{N} \tag{2.4}$$

Let the initial curve be given as

$$\mathbf{C}^i(p) = \sum_{k=0}^{N-1} a_k^i B_k^n(p) = A^i.[B_0^n B_1^n..B_{N-1}^n] \tag{2.5}$$

where a_k are the spline basis coefficients and $B_k^n(p)|_{k=0,1,..N-1} = \beta^n(p-k)$ are the spline basis, n representing the order of the spline. The curve is specified by the user with N node points. Let y represent the curve obtained by evolving our curve according to the curvature flow for one iteration, i.e.

$$y = \mathbf{C}^i(p) + \kappa\, \mathbf{N} dt$$

Now, our task is to project the curve y, into the subspace orthogonally. Let the best approximate of y be represented as $\hat{y} = p_0 B_0^n + p_1 B_1^n + ... + p_{N-1} B_{N-1}^n$. We need to compute the coefficients $p_0, p_1, ..., p_{N-1}$. From the projection theorem we know that the difference vector $(y - \hat{y})$ is orthogonal to the B-spline subspace. Therefore $\langle (y - \hat{y}), B_i^n|_{i=0,..,N-1} \rangle = 0$. These conditions give us N linear equations, which can be written as

$$\begin{pmatrix} \langle y|B_0^n\rangle \\ . \\ . \\ \langle y|B_{N-1}^n\rangle \end{pmatrix} = \begin{pmatrix} \langle B_0^n|B_0^n\rangle & . & . & \langle B_0^n|B_{N-1}^n\rangle \\ . & . & . & . \\ . & . & . & . \\ \langle B_{N-1}^n|B_0^n\rangle & . & . & \langle B_{N-1}^n|B_{N-1}^n\rangle \end{pmatrix} \begin{pmatrix} p_0 \\ . \\ . \\ p_{N-1} \end{pmatrix}$$

i.e.

$$P_B(y) = G.P$$

Therefore

$$P = G^{-1}.P_B(y)$$

The matrix G is known as the *Gram matrix*. The elements of this matrix, for the B-spline subspace are given as

$$G(i,j) = \int_{-\infty}^{\infty} \beta^n(p-i)\beta^n(p-j)\,dp = \int_{-\infty}^{\infty} \beta^n(p-(i-j))\beta^n(p)\,dp$$

G can be computed beforehand, given the number of node points and the order of spline and is a symmetric matrix. $\beta^n(p) = 0$ for $|p| > \frac{n}{2}$. So for computing G, we can take into account the small support of the spline basis functions. If the basis vectors are orthogonal then $G = I$, the identity matrix.
The vector $P_B(y)$ can be computed as

$$P_{B_k}(y)|_{k=0,1\ldots N-1} = \int_0^L y(p)B_k^n(p)|_{k=0,1\ldots N-1}dp$$

where $0 - L$ is the range of the parameter p around the curve. In our case, $B_k^n(p) = \beta^n(p-k)$ and $y(p) = \kappa\mathbf{N}(p)$. The projection gives the change in the spline coefficients of the curve

$$A^{i+1} = A^i + Pdt \tag{2.6}$$

where dt is the time-step. From the new spline coefficients, we can compute the evolved curve at each iteration using Equation 2.5. This method incorporates curvature information in a neighborhood of each node point, thereby giving a better approximation. Although we have explicitly used the spline subspace, this formulation applies any finite linear subspace.
The next step is to decide the time-step in Equation 2.6. We have tried with the well known time-step

$$dt < \frac{1}{max(\kappa\mathbf{N})} \tag{2.7}$$

But, as we will show in the next section, this leads to oscillations in the curve evolution. As we reduce the time-step, the oscillations enter into our evolution at a later time.
In order to overcome this problem, we use a hessian based approach. The curvature flow is a length minimizing flow, which for a spline curve is given as

$$Ł(C(\mathbf{a},p)) = \int_0^1 \sqrt{\left(\sum_k a_{x_k}(B_k^n(p))'\right)^2 + \left(\sum_k a_{y_k}(B_k^n(p))'\right)^2}$$

where $\mathbf{a}$ are the spline coefficients, (a_x, a_y) are the coefficients for the x and y co-ordinates respectively, $(B_k^n(p))'$ are the first order derivatives of the B-spline basis with respect to the parameter p. Computing spline derivatives can be done as shown in [83]. If we minimize this cost function with respect to the spline coefficients $\mathbf{a}$, we get, as expected

$$\left[\frac{\partial Ł}{\partial a_{x_m}}, \frac{\partial Ł}{\partial a_{y_m}}\right] = \int_0^1 \kappa\mathbf{N}[B_m^n(p), B_m^n(p)]dp = \langle \kappa\mathbf{N}, B_m^n \rangle.$$

This is consistent with our projection scheme. The hessian of the cost function with respect to the spline coefficients, is given as

$$Hess(Ł) = \begin{pmatrix} \int_0^1 \frac{C_x'^2(p)B_i'(p)B_j'(p)}{|C'(p)|^3} & \int_0^1 \frac{-C_x'(p)C_y'(p)B_i'(p)B_j'(p)}{|C'(p)|^3} \\ \int_0^1 \frac{-C_x'(p)C_y'(p)B_i'(p)B_j'(p)}{|C'(p)|^3} & \int_0^1 \frac{C_y'^2(p)B_i'(p)B_j'(p)}{|C'(p)|^3} \end{pmatrix}$$

where, $C_y'(p), C_x'(p)$ and $B_j'(p)$ are first derivatives of x and y components of the curve and that of the B-spline basis with respect to the parameter p of the curve and $i, j = 0, 1, ...N-1$. They can be computed as shown in [83]. If the curve has N node points, the Hessian is a $2N \times 2N$ matrix. Let λ_{max} denote the maximum eigenvalue of the hessian matrix. The time-step is given as

$$dt = \frac{1}{|\lambda_{max}|} \tag{2.8}$$

In the Newton scheme, the gradient is multiplied by the inverse of the hessian matrix of the cost function. We take a more conservative approach by using the maximum eigenvalue of the hessian matrix. We take the lowest bound on the step in all directions. This scheme works much better then the one given by Equation 2.7, as can be seen in the results in section 2.4. For numerical computations of all integrations, we use Riemann sums.

2.4 Experiments

First we consider the scheme where we evolve only the node points of the curve according to the curvature and normal at those points, in the B-spline subspace. As one can observe in Figure 2.4, the node points tend to lie on a straight line segment, thereby stopping further evolution. Next we apply the projection algorithm, in which the time-step is computed according to Equation 2.7. We see that oscillations creep into the curve at some stage. We use two different time steps, but still the oscillations creep into the curves. Figure 2.3 is obtained with time step $dt = \frac{1}{2\ max(\kappa \mathbf{N})}$ and Figure 2.4 is obtained with time step $dt = \frac{1}{5\ max(\kappa \mathbf{N})}$. In Figure 2.5, we use the time step given by Equation 2.8. Here the oscillations occur after evolving the curve for a long time as compared to the first scheme. Also, in this case, the oscillations may be because of the fact that the node points on the curve come too close to each other, as discussed in [16]. We show couple of other examples in Figure 2.6.

2.5 Conclusion

We have taken a different view on curve evolution. We approximate the curve evolution PDE in given linear finite dimensional subspaces. This approach is also useful when there is a constraint requiring the curve to remain in a specified subspace. We have given experimental results for the B-spline subspace. The results show that the numerical problems with discretization have been taken care of to a large extent. We also have given a numerical scheme for the time step for the curve evolution in the B-spline subspace,

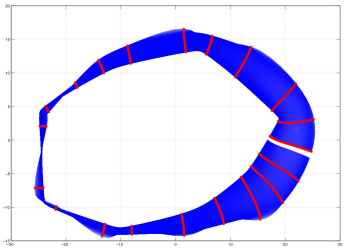


Figure 2.2: Curvature flow, by evolving only the node points of the spline curve. Here the curve is represented by 3^{rd} B-splines, hence the properties of the curve like curvature etc., are accurately computed. But still the evolution takes the node points to lie on a straight line, after which the evolution practically stops. This happens because the curvature information on the entire curve is not taken into account. In our formulation, the projection operator takes into account the curvature information in the neighborhood of the node points, as shown in the previous section

which is a variant of the Newton scheme. Further theoretical study regarding the stability of the scheme needs to be done. Currently the formulation applies to linear subspaces, but it should be possible to extend it to non-linear subspaces also.

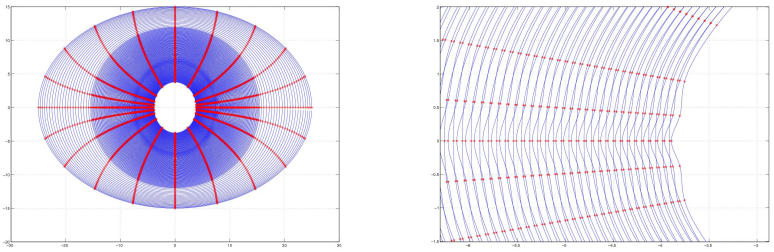


Figure 2.3: (left) Curvature flow of points on an ellipse, using time step given in Equation 2.7. (right) Magnified view of the figure on left, note the oscillations in the curve

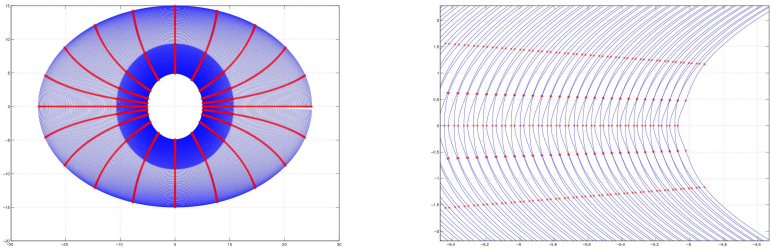


Figure 2.4: (left) Curvature flow of points on an ellipse, using time step given in Equation 2.7. (right) Magnified view of the figure on left, note the oscillations. Here the time step is still smaller then the one used in Figure 2.3

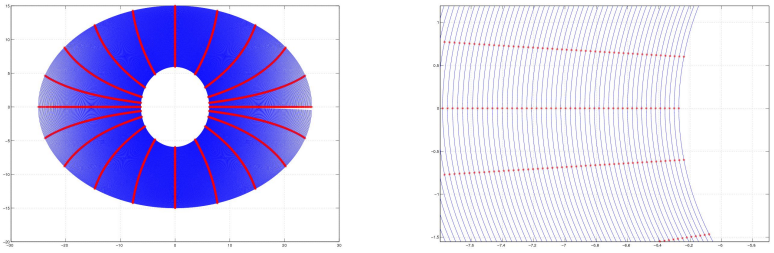


Figure 2.5: (left) Curvature flow of points on an ellipse, using time step given in Equation 2.8. (right) Magnified view of the figure on left, note that there are no oscillations

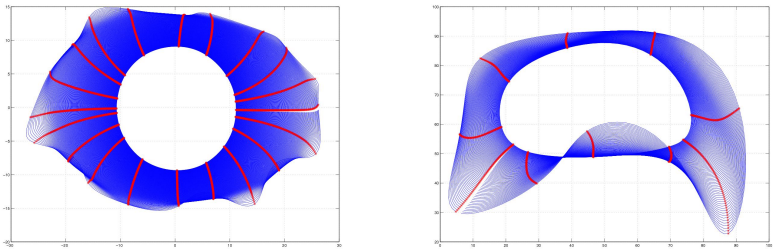


Figure 2.6: (left) Curvature flow of a curve, using our approach, using time step given in Equation 2.8. (right) Curvature flow of a curve, using our approach, using time step given in Equation 2.8. These are couple of examples of curves having convex and concave parts

Chapter 3

Bicycle chain shape manifolds

This chapter has been published as *Bicycle Chain Shape Models* in proceedings of the *CVPR Workshop on Mathematical Methods in Biomedical Image Analysis, 2009*.

3.1 Introduction

There is a wide literature on shape representation and shape analysis in Computer Vision and Medical Imaging as shape understanding is one of the most fundamental task in Image Analysis. A 2-dimensional shape is generally defined as an equivalence class of smooth 1-dimensional submanifolds of $\mathbb{R}^2$ modulo similarity [51]. Computational representations, ranging from the simplest to the most sophisticated, have been suggested in the past, e.g. point set distributions [42, 5], linear point distribution models (PDM) [15], parametric representations via B-splines, levelset representations [71], and their adaptation, as for example, specific shape constraints, soft priors, etc..., for an ever growing amount of tasks.

Manual annotations of anatomical structures in medical images, such as X-rays, Ultra Sound, are routinely performed by radiologists and other experts in many clinical studies, resulting in the encoding of shapes as point set distributions. Point set distributions for shape representation and analysis are of tremendous importance in Medical Imaging. Deriving such distributions presupposes consistent annotations, which is not always the case: the following figure shows two annotated vertebra shapes from X-ray images, during a clinical study on vertebra fractures, the first vertebra is annotated with 31 points, the second with 32. Moreover the number of points between corner landmarks (the circular ones) do not match for corresponding pairs. This is caused by the absence of clear ground truth landmarks along the endplates of the vertebrae. In order to tackle this somehow common situation, a resampling is necessary; pseudo-landmarks should be placed such that the resulting model is more compact, no additional variation caused by points sliding along the outline should be modelled. Some recent approaches for curves and surfaces were proposed for instance by Davies *et al.*. [18] using minimum description length to solve this problem, while, for surfaces, Cates *et al.* used an entropy based particle system approach in [10].

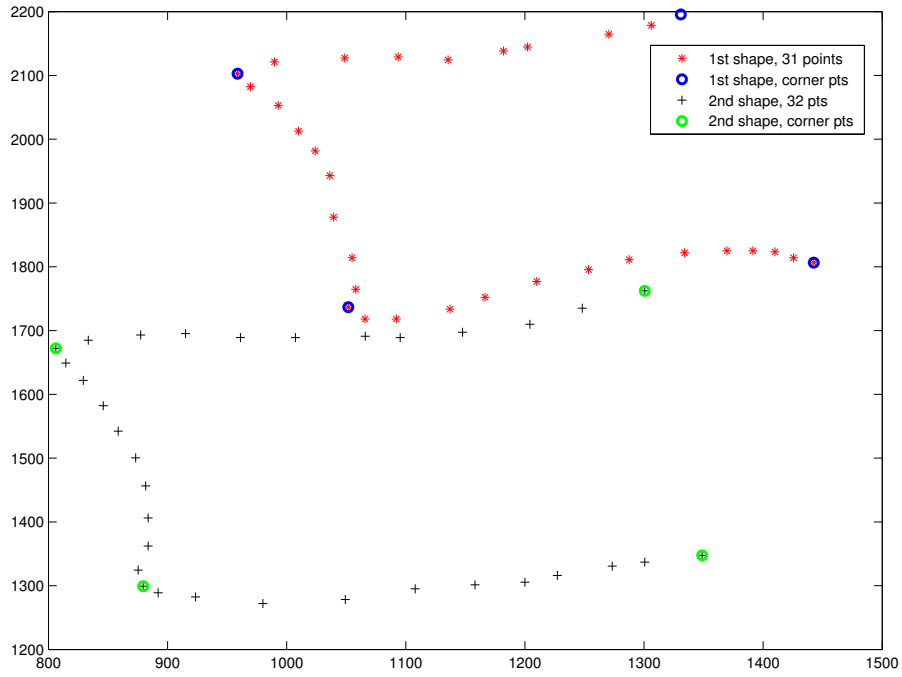


Figure 3.1: Two annotated vertebrae from a clinical study. The number of annotation points differ.

For curves, which are the objects of interest in this paper, a way to do it is to first impose a fixed number of pseudo-landmarks between landmarks, regularly distributed along the outline between the landmarks. This regularity often takes the form of an equidistance constraint for pseudo-landmarks situated between consecutive landmarks. This can be formulated as setting the variance of the distribution of planar distances (or square-distances) between consecutive pseudo-landmarks to 0. In a figurative way, a segment between two consecutive landmarks is similar to a segment of a **bicycle chain**, for the links that constitute a bicycle chain have the same length! This has the nice property of minimizing the variability due to the annotation process. But once this resampling has been performed, forgetting this variance constraint induces apparent extra variability which may be difficult to handle due to the non linearity of the constraint. This is illustrated in Figure 3.2 where the Euclidean mean of the upper and lower curves does not have equidistant pseudo-landmarks introducing extra variability on the horizontal placement of the pseudo-landmarks. We propose to handle this situation by introducing the constraint explicitly in the descriptions of our preshape spaces. This null-variance can be reformulated as a series of simple quadratic constraints on the pseudo-landmarks and will, for shapes determined by n points in $\mathbb{R}^d$, define implicitly a submanifold of the point set spaces $(\mathbb{R}^d)^n$. Endowed with the metric that comes from the standard Euclidean structure of $(\mathbb{R}^d)^n$, it becomes a Riemannian manifold. In order to obtain point distributions models, Generalized Procrustes Analysis (GPA) [28] should be performed with the

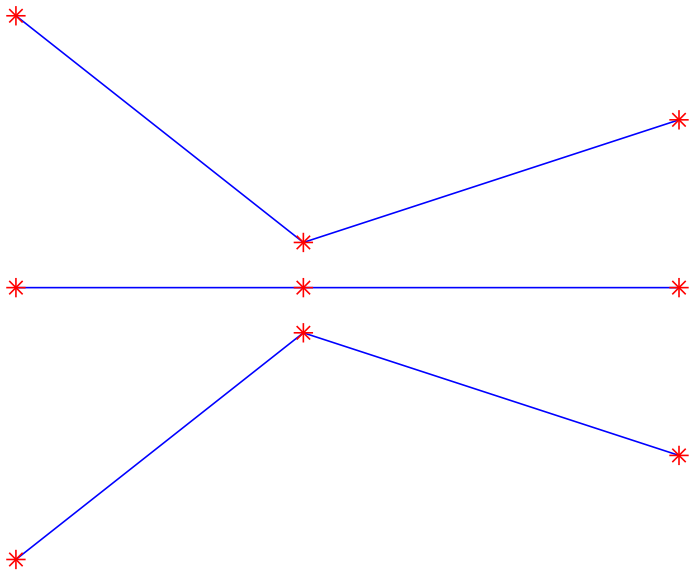


Figure 3.2: Two 3-point curves and the Euclidean mean.

induced metric, leading to what we will call Geodesic Generalized Procrustes Analysis (GGPA), while Principal Component Analysis should be replaced by Principal Geodesic Analysis (PGA) [24] in order to take into account the curved structure of the manifold. In the rest of this paper, we will focus to point set configurations in $\mathbb{R}^2$. This will simplify the presentation. Extension to 3D curves can be carried out easily.

So as to be able to compute GGPA and PGA, we need tools for computing Riemannian exponential map, geodesics, and log map on implicitly defined submanifolds. By extending computations of exponential map to provide not only geodesic, but corresponding moving frames, we propose a shooting method for computing Log maps. When it fails, we replace it by a path straightening algorithm based on local properties of geodesics.

This paper is organized as follows. In the next section we introduce the preshape manifolds that we use as well as the geometric tools needed for our statistical analysis: Geodesic Generalized Procrustes analysis and Principal Geodesic Analysis. Exponential and Log maps are discussed in Section 3.3. In Section 3.4 we present experiments; the first one on the 3-points toy example and the second on a data set of vertebra coming from a clinical study on vertebra fractures. Finally we conclude in Section 3.5.

3.2 Preshape manifolds

In point based models, a typical object consists of q landmark points and n_k, $k = 1, ..., q-1$ ($k = 1, ..., q$, for closed configurations) pseudo landmarks between consecutive landmark points. A segment of this object consists of $n_k + 2$ points, n_k pseudo-landmarks $P_i, i = 2, ..., n_k + 1$ between 2 landmark endpoints P_1, P_{n_k+2}. The objects we consider consist of such configurations with equal (squared) Euclidean distance between the neighboring points in each of the segments. This characteristic distance will generally vary from segment to segment and objects to objects, even when the sequence of numbers $(q, n_1, \ldots, n_{q-1})$ is fixed. We start by describing constraints on segments.

3.2.1 n-Links Bicycle Chain Manifolds

Here onwards we work on one segment with $n_k = n - 2$ pseudo-landmarks between 2 landmark endpoints. Then the equidistant constraint can be written as a simple quadratic constraint $F : \mathbb{R}^{2n} \to \mathbb{R}^{n-2}$ given as

$$F_i(P_1, ..., P_n) = d_{i+2,i+1} - d_{i+1,i}, \;\; i = 1, .., n-2 \tag{3.1}$$

where $d_{i,j} = (x_i - x_j)^2 + (y_i - y_j)^2$ is the squared euclidean distance between points P_i and P_j, The configuration space is the subspace of $\mathbb{R}^{2n}$ given by $A_n = F^{-1}(0) \backslash \Delta$, where Δ is the "diagonal" $\Delta = (P, \ldots, P) \in (\mathbb{R}^2)^n$ consisting of segments reduced to a single point, for, while $\Delta \subset F^{-1}(0)$, the rank of F breaks down exactly along Δ. This ensures that A_n is a submanifold of $(\mathbb{R}^2)^n = \mathbb{R}^{2n}$ [6] The tangent space of A_n at a segment x is given by

$$T_x A_n = \ker(JF(x))$$

the kernel (or null space) of the Jacobian of F at point $x \in A_n \subset \mathbb{R}^{2n}$. By restricting the scalar product of $\mathbb{R}^{2n}$ to $T_x A_n$, A_n is endowed with a Riemannian Metric [6]. We may call A_n a **n-links bicycle chain segment manifold**.

More general point configurations are then built by concatenating these n-links bicycle chain segments, imposing endpoint matching which are linear constraints. When the number q of landmarks points and the numbers n_k, $k = 1, \ldots, q-$ of pseudo-landmarks points are fixed, corresponding configurations form a Riemannian submanifold of the product manifold $\prod_{k=1}^{q} A_{n_k+2}$, and this manifold has also the metric inherited from the embedding space $(\mathbb{R}^2)^N$ with $N = q + \sum_{k=1}^{q-1} n_k$.

Having a Riemannian metric, we can compute length of paths in these manifolds, define geodesic and geodesic distances [6].

3.2.2 Removing Translation and Scaling

In the following, we denote by $\mathcal{M}$ such a configuration manifold. To work with preshapes in the sense of [42], we need to quotient out translations and scaling from points in $\mathcal{M}$ (although in some applications, scale could be an important feature of the shape). Removing translations is as usual easy. If $\mathcal{M}'$ denotes

the submanifold of $\mathcal{M}$ of configurations with centroid at the origin of $\mathbb{R}^2$ then $\mathcal{M} \simeq \mathcal{M}' \times \mathbb{R}^2$, by sending a configuration $S = (S_1, \ldots, S_n)$ to $(S - \bar{S}, \bar{S}$ where $\bar{S} = \frac{1}{n}\sum_{i=1}^{n} S_i$ is the centroid of S. This decomposes $\mathcal{M}$ into two *orthogonal* factors, which imply that a geodesic path in $\mathcal{M}$ between centered objects in $\mathcal{M}'$ will be in fact a geodesic path in $\mathcal{M}'$. From now on we therefore assume that all our configurations have centroid at $0 \in \mathbb{R}^2$. Following [42], we remove scale by imposing $\|S\|^2 = \sum_{i=1}^{n} \|S_i\|^2 = 1$, i.e by intersecting $\mathcal{M}'$ with the unit sphere of the embedding space. This defines a new submanifold $\mathcal{S}$ of $\mathcal{M}'$, and $\mathcal{S}$ is our preshape manifold.

3.2.3 Geodesic Generalized Procrustes Analysis

Given a sample set $(S_i)_{i=1\ldots n} \in \mathcal{S}$, our GPA follows [42], but is performed on $\mathcal{S}$. It attempts to compute a set of planar rotations $R_{\bar{\theta}_i}$, $i = 1, \ldots, n$ and a preshape $\bar{\mu} \in \mathcal{S}$ minimizing the misalignment criterion

$$E(\theta_1, \ldots, \theta_n, \mu) = \sum_{i=1}^{n} d(R_{\theta_i} S_i, \mu)^2. \tag{3.2}$$

where d is the geodesic distance in $\mathcal{S}$. This will result in an aligned preshape sample $(\bar{S}_i := R_{\bar{\theta}_i} S_i)_{i=1\ldots n}$, $\bar{\mu}$ being the Fréchet mean ([43]) of this sample and the distances $d(\bar{S}_i, \bar{\mu})$ should represent the true *shape* distances to this mean.

The minimization procedure for (3.2) is sketched in Algorithm 1. We describe briefly the loop steps. A first guess for the rotations is computed by standard Euclidean rigid registration [28] providing candidate rotation angles for each shape. Then we search for the rotations angles that minimize the true geodesic distances in a neighborhood of the previously obtained angles. The Fréchet mean is then computed by adapting the procedure described in [24] to our case. In Figure 3.3 the need for the minimization search after the initial Euclidean registration is illustrated by showing a base preshape, and rotation of a second preshape with respect to Euclidean and submanifold distances.

Algorithm 1 Calculate the mean shape μ' and the aligned shapes S_i''

Require: $S_i \in \mathcal{S}, i = 1, ..., m$
  $\mu' = S_1$ {initial guess}
  repeat
    Set $\mu = \mu'$. $S_i' = S_i$ shapes aligned to μ using Euclidean distances.
    $S_i'' = S_i'$ shapes aligned to μ using *geodesic* distances.
    $\mu' =$ Fréchet mean of (S_i'')
  until $d(\mu, \mu') <$ Threshold.
  Output: Mean μ', aligned shapes S_i''.

3.2.4 Principle Geodesic Analysis (PGA)

PGA is a generalization of Principal Component Analysis (PCA) to nonlinear manifolds [24]. We seek to compute a minimum number of tangent vectors at the mean, which generate geodesics that represent as much variability in the data on the manifold as possible. Thus PGA is PCA done on the tangent space

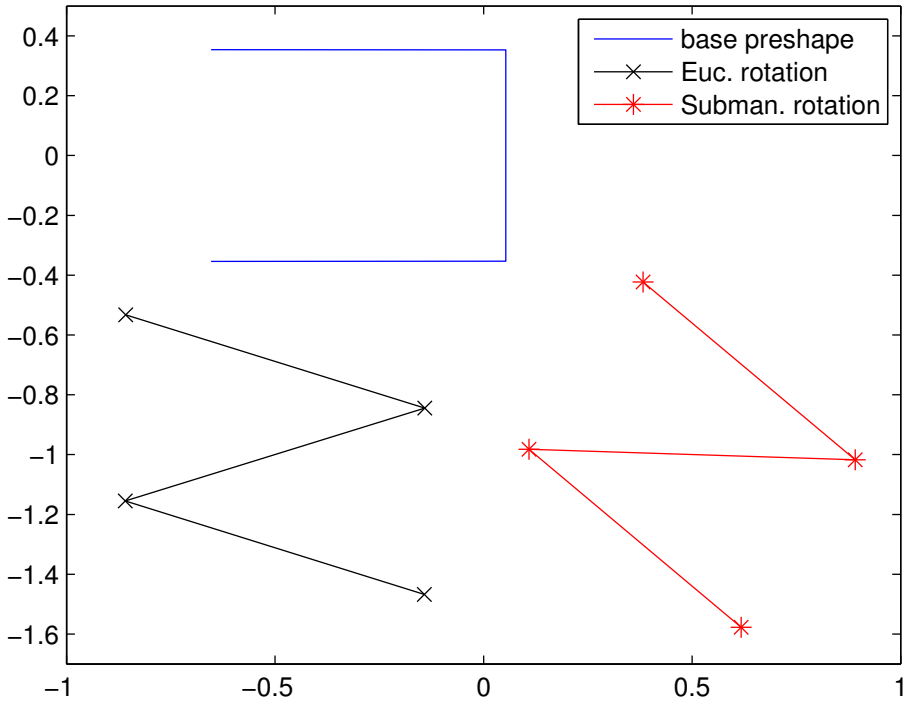


Figure 3.3: A base preshape and Euclidean and manifold registration.

of the mean. Unfolding the manifold to this tangent space is performed by the Riemannian Log map. The algorithm can be summarized as:

- Given m preshapes in $\mathcal{S}$ and the mean preshape μ, compute $v_i = Log_\mu(S_i)$, $i = 1, ..., m$, the tangent vectors for each preshape in the tangent space at the mean.
- Compute the covariance matrix $C = \frac{1}{m}\sum_{i=1}^{m} v_i v_i^T$
- Compute the eigenvectors and eigenvalues (e_i, λ_i) of C.

The geodesic paths corresponding to the tangent vectors $e_i \in T_\mu\mathcal{S}$ are the principle geodesic components.

Computing the Fréchet mean and PGA use Exponential map, Log map and geodesics on implicitly defined Riemannian manifolds. They are described in the next section

3.3 Geodesics on the manifold; the Exp- and Log-map

Geodesics are fundamental to the theory of Riemannian manifolds ([6]). They are closely related to the Exponential map $\mathrm{Exp} : TM \to M$ in the sense that a geodesic γ trough the point p with initial velocity

vector v is given by the curve

$$\gamma(t) = \mathrm{Exp}_p tv \ .$$

The map Exp_p is invertible in a sufficiently small neighborhood of 0 in T_pM. When U is such a neighborhood we denote by $\mathrm{Log}_p : \mathrm{Exp}_p(U) \to U$ the inverse of Exp_p.

The distance between two elements of the manifold is given by

$$d_M(p,q) = \inf \left\{ l(c) \middle| c \text{ is a curve joining } p \text{ and } q \right\} \ .$$

Here $l(c)$ denotes the length of the curve c. Since geodesics are critical points of the length functional, it is in the case of a complete manifold M sufficient to consider geodesics when computing the distance. Therefore, if p and q are sufficiently close so that only one geodesic joins them,

$$d_M(p,q) = \|\mathrm{Log}_p q\| \ . \tag{3.3}$$

In general we cannot be sure that a given geodesic joining p and q is length minimizing. In such cases, we define $\mathrm{Log}_p q$ to be the initial direction of *some* geodesic joining p and q and use (3.3) as a guess on the distance.

Computing Exp_p amounts to solving an initial value ODE problem. This can be done neatly numerically, confer [19]. Computing $\mathrm{Log}_p q$ is substantially harder. We make use of a shooting method ([46], [56]) for computation of Log_p for input values close to p, and a path-straightening method for non-local input.

3.3.1 Shooting method

A shooting method iteratively improves an initial guess by repeatedly computing a residue or error correction, and updates the initial guess using that. Based on the fact that Log_p is the inverse of Exp_p, our basic algorithm is presented in Algorithm 2. The ability to compute length and direction in Euclidean

Algorithm 2 Calculate $v = \mathrm{Log}_p q$ on $\mathcal{S}$ by shooting

Require: $p, q \in \mathcal{S}$
  $v \Leftarrow$ projection of $q - p$ to $T_p\mathcal{S}$ {initial guess}
  repeat
    $\tilde{q} \Leftarrow \mathrm{Exp}_p v$ {shot based on guess}
    $\tilde{r} \Leftarrow$ projection of $q - \tilde{q}$ to $T_{\tilde{q}}\mathcal{S}$ {residue at $\tilde{q}$}
    $r \Leftarrow$ par. transport of $\tilde{r}$ to $T_p\mathcal{S}$ {residue at p}
    $v \Leftarrow v + r$ {update v}
  until $\|\tilde{q} - q\|_{\mathbb{R}^{2n}}$ is sufficiently small.

space and the implicit representation of $\mathcal{S}$ as a submanifold of Euclidean space enables us to compute both the initial guess, update v, and compute the Euclidean error of our guess. When q is close to p these estimates work well and improve the situation in [56] where the embedding space approximations are not at hand and e.g. the update of v therefore is based on numerical estimates of the gradient of a cost

functional.

We use the projection of the vector $q - p$ in embedding Euclidean space to the tangent space $T_p\mathcal{S}$ as our initial guess. In each iteration we compute $\mathrm{Exp}_p v$ and express the error by the Euclidean distance $\|q - \mathrm{Exp}_p v\|$. We update v by projecting the Euclidean residue $q - \mathrm{Exp}_p v$ onto the tangent space $T_{\mathrm{Exp}_p v}\mathcal{S}$, parallel transport the resulting vector to $T_p\mathcal{S}$ and add it to v; this procedure is the natural manifold generalization of error correction in Euclidean space.

The parallel transport is computed using a parallel frame along the curve $t \mapsto \mathrm{Exp}_p tv$. We compute the parallel frame by using the fact that parallel vector fields have zero intrinsic acceleration, introduce a Lagrange multiplier, and solve the resulting ODE. The computation of the frame can be nicely coupled with the computation of $\mathrm{Exp}_p v$ when using the method of [19].

The shooting method relies completely on the quality of the initial guess and updating residues. Both are determined by how well the projections on the tangent spaces approximate the paths on the manifold, or, in other words, how close to linear the manifold is; in an Euclidean manifold the shooting method converges in one iteration whether as it on a torus might not converge at all. It will though always converge locally due to the smoothness of our manifold.

An additional drawback of the shooting method is its sensitivity to numerical errors in the computation of Exp_p. This can especially be a problem if the curvature around the target element q is large, confer [45].

3.3.2 Path straightening

When the shooting method fails to converge due to large curvature of the manifold, we apply the path straightening method of [57]; we update an initial curve by repeatedly shooting between pairs of points on the initial curve close to each other. The closeness assures the convergence of the shooting method. In each iteration the curve is a piecewise geodesic and by repeatedly changing the points between which we shoot, the non-smooth bends of the curve are removed. Since geodesics are critical points of the length functional, we stop the process when we get no signification reduction of length on each iteration.

Path straightening requires an initial path. In practice we get this path by shooting until we detect non-convergence of the shooting method. We then restart the shooting method with the best guess from the previous run as our new starting point. In practice we always obtain convergence of the shooting method in the second run. Now concatenating the geodesics obtained from the two runs gives a piecewise geodesic connecting the points which can serve as input to the path straightening algorithm. In case this method fails, we explicitly make an initial path.

As noted in [57] we may need to extract a subsequence in order for the path straightening algorithm to convergence to a geodesic. In practice we do not experience such situations, and we accept the possibility of this happening in the same way as we accept that geodesics might not be length minimizing.

3.4 Experiments

We present two examples illustrating the effect of our manifold setting. We start by discussing the dimensionality reduction gained in a small 3-point example and then progress to study a dataset of vertebrae shapes.

3.4.1 Illustrative example

In Figure 3.4 we see three 3-point preshapes with equidistant points. They are all normalized and hence reside on the manifold $\mathcal{S}$. The middle preshape is the Fréchet mean of the upper and lower preshapes, and hence the mean of all three preshapes.

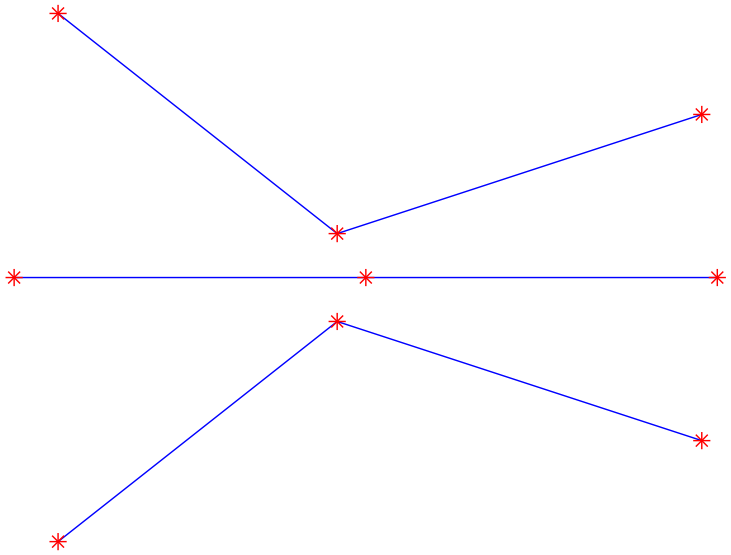


Figure 3.4: Three 3-point preshapes on the manifold.

The manifold $\mathcal{S}$ has two dimensions. Doing a Principal Geodesic Analysis on the set of the three preshapes, we get one mode of variation. The geodesic corresponding to this mode connects the three preshapes as illustrated in Figure 3.5. Note that in the figure the preshapes have been placed in the plot as to have zero mean.

Now suppose we disregard our manifold notion and attempt to do Euclidean Principal Component Analysis in the embedding Euclidean space. The Euclidean mean of the three preshapes will again be a straight line, but in this situation the points on the mean will not be equally spaced and hence the mean will not be in $\mathcal{S}$. When computing the PCA we get two modes of variation; one mode representing vertical motion as illustrated in Figure 3.6, and one mode representing horizontal motion. The latter mode arises from the placement of the points on the straight line mean and is thus irrelevant. Therefore, in this example the PCA captures only 97.5 percent of the variation in a mode giving relevant information. This

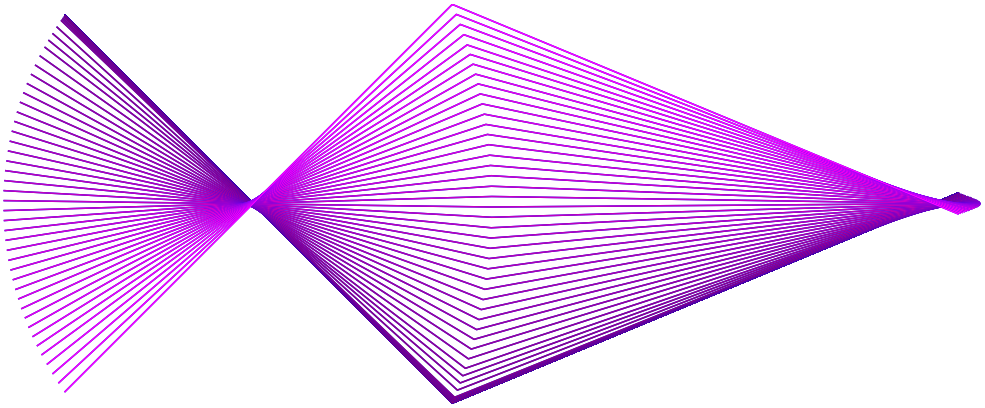


Figure 3.5: The geodesic corresponding to the only mode of variation obtained from PGA.

contrasts that PGA captures all variation.

3.4.2 Vertebra shapes

Our dataset of vertebrae consists of 304 manually annotated vertebra shapes on lateral X-rays. For each vertebra, outlines have been manually drawn by choosing points along the contours, assuming a simple linear interpolation between them. Corner points of the vertebra endplates are indicated but do not always match the outlines perfectly. New corner points have been defined as the points of the contour that are closest to the manually annotated corners. This divides the outline into 3 segments, the upper, left and lower ones. For resampling, we fixed the number of pseudo-landmarks per segment to 16, leading to 52 points per shape. The pseudo-landmarks positions were computed segmentwise so as to minimize a squared-distance between the original outline and the new one. Given an n-tuple $\mathcal{P} = (P_1, \ldots, P_n)$ of equidistant-spaced points, with P_1 and P_n being the *fixed* corner points of this segment, let $C_{\mathcal{P}}(t)$ be the piecewise linear curve joining them, and $C_0(t)$ the piecewise linear curve formed by joining the original annotated points for the corresponding segment. We minimize the squared-distance

$$E(\mathcal{P}) = \int (C_{\mathcal{P}} - C_o)^2 \, dt.$$

We start with a configuration $\mathcal{P}$ on the straigth line segment joining P_1 to P_n and perform gradient descent on the corresponding preshape manifold $\mathcal{S}$ using the exponential map. The result of applying the redistancing procedure to the manually annotated vertebra in Figure 3.7 is shown in Figure 3.8.

In our illustrative example it is clear that we introduce non-linearity when restricting to the manifold. In order to illustrate that we have significant curvature also in the relatively high dimensional manifold

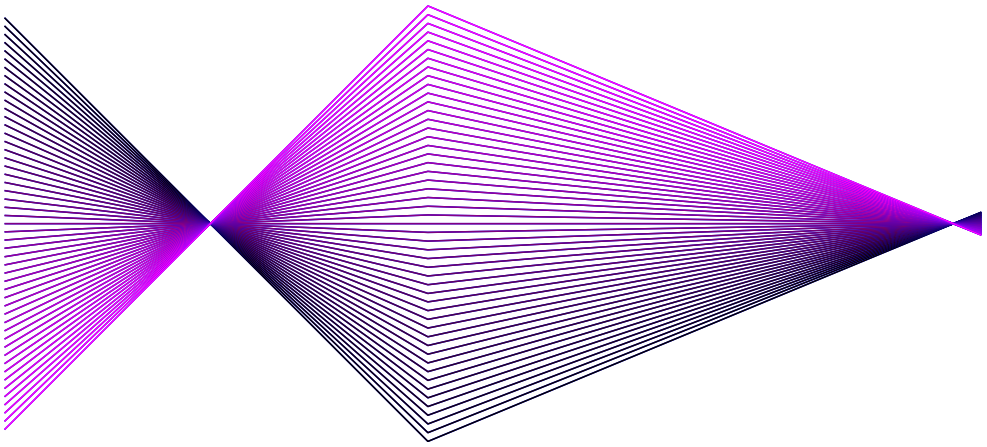


Figure 3.6: One of two modes of variation for PCA.

used for the vertebrae, we compute the Fréchet mean vertebra and measure an approximate distance from each vertebra to the tangent space of the mean; we let v_m denote the mean and for each vertebra v we compute $w = \mathrm{Log}_{v_m} v$. We then let x be the distance $\|v - (v_m + w)\|$ between the vertebra and an approximated projection to $T_{v_m}\mathcal{S}$, and record the relative distance $x/\|w\|$. A non-curved manifold would result in zero relative distance. We see a mean relative distance of 12 percent clearly indicating that the manifold is curved. Performing the same computation on the non-normalized manifold $\mathcal{M}'$ gives a mean relative distance of 9 percent indicating that not all curvature arises from the normalization to the unit sphere.

Figure 3.9 illustrates how PGA provides a more compact description than PCA. The figure shows the normalized sum of the first n eigenvalues as a function of n. It can be seen that in order to capture say 99.5 percent of the variation, we will need 25 eigenvectors when doing PCA as opposed to only 20 eigenvectors when doing PGA.

3.5 Conclusion

In this paper we have introduced manifolds of preshapes built by constraining distributions of pseudo-landmarks between pairs of consecutive landmarks. This endows these preshape manifolds with a structure of Riemannian manifolds. We have developed tools for computing Exponential maps, Log maps, geodesic distances, allowing us to define a Geodesic GPA and adapt PGA to that situation. We have shown on examples that PGA captures variability better that PCA.

Although we have built our models for planar point configurations, they are clearly not restricted to this case. Other types of length and position constraints can also be used. We are also not restricted to shape manifolds. The techniques presented in this work can be used to perform statistics on other

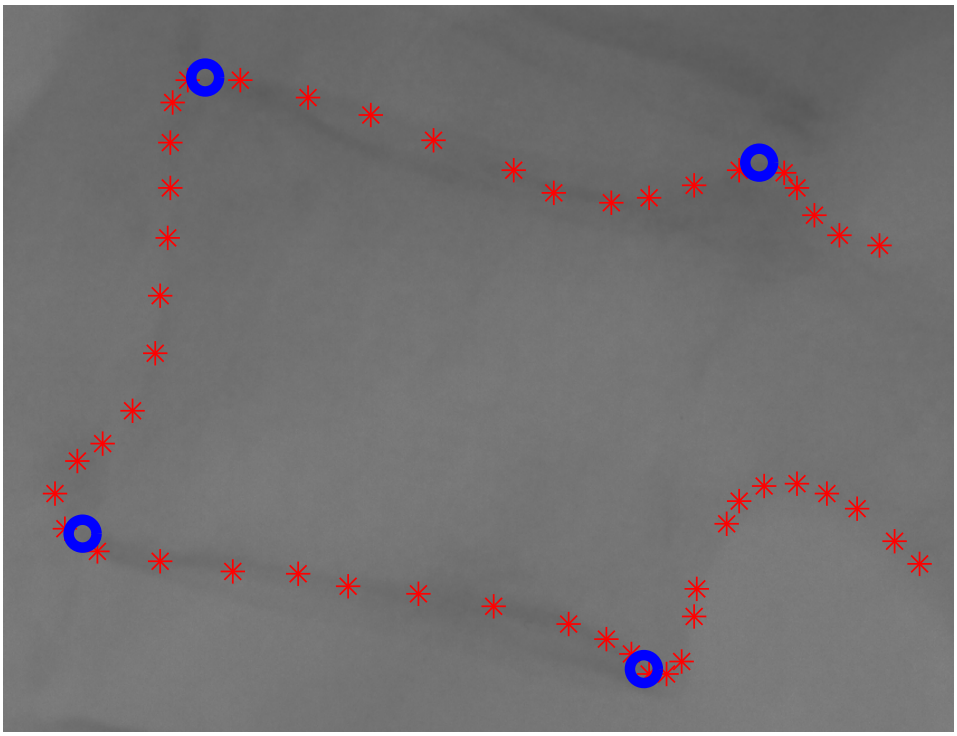


Figure 3.7: Manually annotated vertebra.

submanifolds of a linear configuration space implicitly defined by a set of smooth constraints. This is the subject of ongoing work.

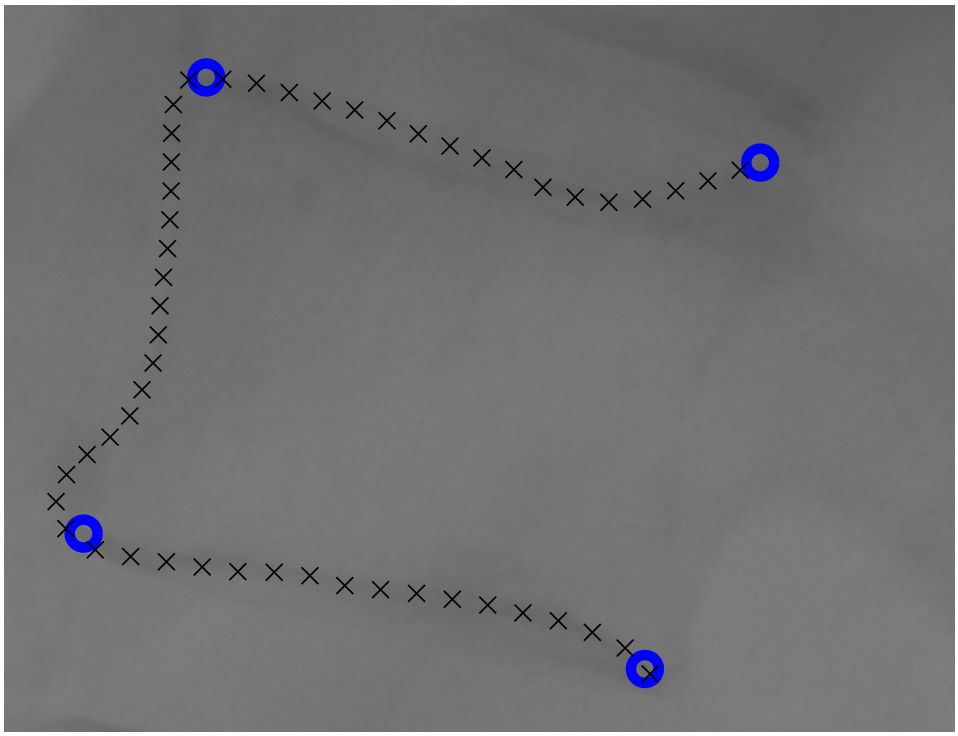


Figure 3.8: Result of applying redistancing procedure.

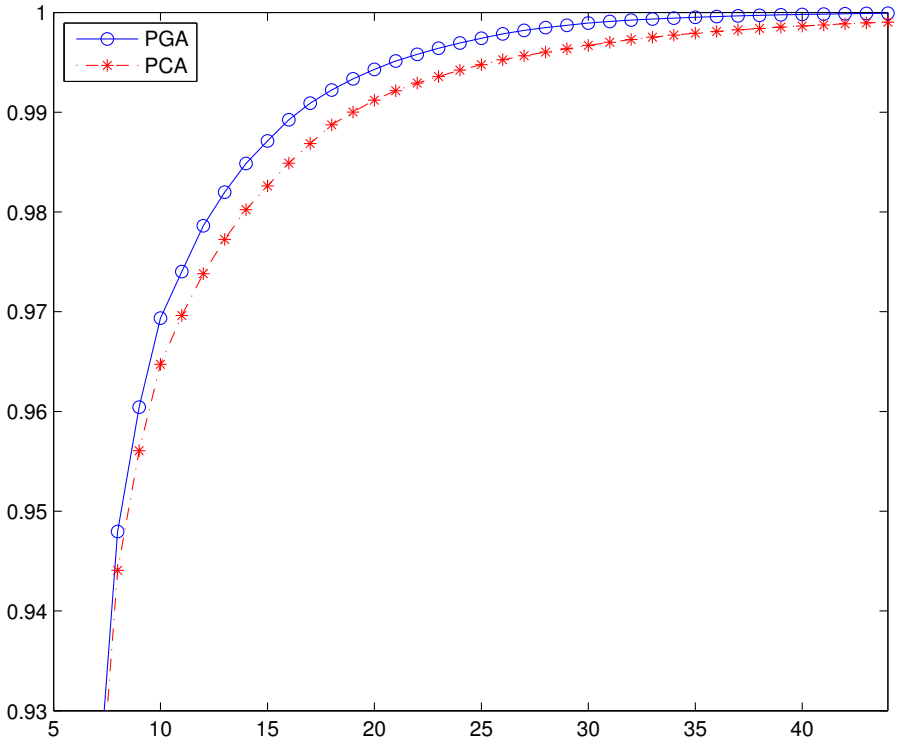


Figure 3.9: Accumulated spectrum of PGA and PCA.

Chapter 4

Curve Evolution in Implicitly defined Curved Subspaces

4.1 Introduction

Curve evolution has now become a standard tool in Computer Vision. It has been used for tracking interfaces, registration, active contour algorithms for segmentation, etc. [38, 72, 44, 3]. Curve evolution involves deformation or motion of a curve via a given velocity vector field defined on a space of curves that share some common structures.

Curve evolution equations arise generally either as a gradient descent of a cost function or a 'direct design' approach where the curve evolves along a velocity vector field which is defined so as to meet some specific requirements. How to manipulate the curve objects?

Curves can be manipulated implicitly as the zero-level set of a given function for instance, or explicitly via a parametrization. In this paper we are interested in the latter so care must be taken to understand the effect of evolution on the parameterization of the curve. We evolve a curve in a geometric way which ensures parametrization independence and does truly evolve the underlying 1-dimensional submanifolds of the plane. A general reference for the mathematical treatment of shape related problems and their representation is the monograph of Delfour and Zolésio [20].

When curve evolution is simulated on a computer, due to the inevitable discretization involved, the evolution may not follow the theoretical flow intended by the equation, and is carried out in some finite dimensional subspace of the generic space of curves. Moreover technical considerations may lead to further restrict the manifold of curves in which the evolution takes place, for instance by imposing specific constraints on the curve representations. We deal specifically with implicit constraints, which could generate a curved subspace. This paper presents numerical scheme to perform the evolution in these settings.

This chapter is organized as follows. In the next section we give some background of our work and discuss some related work. In section 4.3 we present a general formulation of the problem and discuss

associated projection schemes. In section 4.4, we explain the heirarchy of the subspace of curves that we are interested in alongwith the inner products defined on them. We explain the projection steps in detail in section 4.5. In section 4.6 we discuss the resulting scheme when curves are represented by coordinates in finite dimensional numerical spaces. In Section 4.7 we take up the example of B-spline curves with equidistant node points in the plane - the N-links bicycle chain space. We discuss how to restrict the geodesic active contours to the N -links bicycle chain space in Section 4.8 and present a series of experiments in Section 4.9. We give some concluding remarks and future work directions in section 4.10.

4.2 Background & Related work

Curve evolution is given by the following partial differential equation:

$$\frac{\partial C}{\partial t}(s,t) = \mathbf{V}(s,t)$$

where C is a curve with parameter s, V is the velocity vector at point $C(s,t)$. The velocity vector can be decomposed into the tangential and normal component:

$$\mathbf{V} = V_n\mathbf{n} + V_t\mathbf{t}$$

where $\mathbf{n}(s,t)$ and $\mathbf{t}(s,t)$ are the normal and tangent to the curve at point $C(s,t)$. The tangential component of the velocity affects just the parameterization of a curve [23], so the geometric curve evolution equation is given as,

$$\frac{\partial C}{\partial t}(s,t) = V(s,t)\mathbf{n}(s,t)$$

where V is the scalar velocity in the normal direction. The velocity can be obtained in two ways. One is the direct design approach where the application in hand dictates the evolution of a curve and therefore specifies the velocity directly. Using the gradient descent approach to minimize energy functionals on curves is another way to compute the velocity. For example, consider the following energy functionals:

1. Length of the curve:

   $$\mathcal{L}(C) = \int_0^p |C'(s)|ds$$

   The gradient descent equation gives the classical Curvature flow [31]:

   $$\frac{\partial C}{\partial t} = -\frac{\partial \mathcal{L}}{\partial C} = \kappa\mathbf{n}$$

   where κ is the curvature of the curve.

2. Geodesic active contour energy:
   Geodesic active contours are used for segmentation of object boundaries [9]. The energy of a curve

is defined as:

$$\mathcal{E}(C) = \int_0^p g(|\nabla I(C(q))|)|C'(q)|\, dq$$

where $I(x)$ is the image and $g : [0, +\infty[\rightarrow \mathbb{R}_+$ is a strictly decreasing function with $g(r) \rightarrow 0$ as $r \rightarrow +\infty$. The gradient descent equation is given by

$$\frac{\partial C}{\partial t} = -\frac{\partial \mathcal{E}}{\partial C} = g(I)\kappa \mathbf{n} - (\nabla g \,.\, \mathbf{n})\mathbf{n}$$

3. Sum of Squared distances between curves:
   The intrinsic mean curve [60, 24] of a collection of curves is given by the minimizer of the sum-of-squared distance function:

$$\mathcal{D}(C) = \frac{1}{2N} \sum_{i=1}^{N} d(C, C_i)$$

   where $C_i, i = 1, 2..., N$ are the given curves whose mean has to be computed. The distance $d(C, C_i)$ is the geodesic distance between curves C and C_i on the manifold of curves considered. The gradient descent equation [37] is given by:

$$\frac{\partial C}{\partial t} = -\frac{\partial \mathcal{D}}{\partial C} = \frac{1}{N} \sum_{i=1}^{N} Log_C(C_i)$$

   where $Log_C(C_i)$ is the logarithm map on the given manifold of curves at the curve C of the curve C_i

In general, the family of curves $C(., t), t \in [0, \infty)$ obtained from curve evolution PDE's form a path of curves in some infinite dimensional space of curves. Our contribution is to restrict this family of curves to a given finite dimensional curved subspace of curves.

Let us mention a few examples of work involving restriction of the evolution space. In [66], Sapiro and Tannembaum restrict the curves under curvature flow to the space of curves with constant areas. In [16], Cremers *et al.* restrict the evolution space to the finite dimensional space spanned by some example shapes. This is used as statistical priors for active contour algorithms to aid segmentation. In [85], Wang *et al.* constrain active contours to a subspace having desired regional properties. Srivastava *et al.* [77] restrict curves to a subspace of curves with finite number of fourier coefficients. In [80] Tatu *et al.* restrict the curve evolution to finite dimensional linear spaces. We extend the formulation to finite dimensional nonlinear spaces as constraints arising in many problems are often non linear.

The ideas presented here bear some similarity with the work of Fletcher *et al.* in ([24]) on shape statistics where shape manifolds are non linear. It also bears some similarity with the work of Hairer ([33]) in the field of numerical mathematics, where the authors solve ordinary differential methods on manifolds numerically.

Since we work on subspaces of infinite dimensional space of curves, it is important to look at examples of other subspaces of curves (and their structure) in the literature. Michor and Mumford [51] work on space of smooth embeddings with reparameterizations being identified. They show that the geodesic distance between any two curves in the space is zero using the standard H^0 between normal deformations of a curve $c(s)$:

$$\langle h, k \rangle_{H^0} = \int_0^L h(s)k(s)|c'(s)|ds.$$

This happens because of wildly oscillating deformations of a curve. In example 3 above, we cannot compute the mean of a collection of curves using the H^0 metric since the distance between any two curves is zero. To overcome this problem, there are primarily two methods. One is to modify the metric on the space and the other is to work on a suitable subspace of this infinite dimensional manifold of curves thereby avoiding the wild oscillations. In essence, in one way or the other, regularity is introduced in the space of curves. Michor and Mumford propose a curvature dependant metric in [51]. In [78], the authors use a Sobolev metric on the space of smooth embeddings. The metric penalizes derivatives of the deformation fields, thus limiting the oscillating deformations of a curve. Klassen et. al. [77] use the second methodology of restricting to some subspace. They use finite number of fourier descriptors to represent the deformations of a curve, thus limiting the frequency of oscillating deformations. Charpiat et. al. [11] consider the space of shapes represented by distance functions with bounded curvature and use a Hausdorff kind of a metric on this space. Due to the bounded curvature constraint the oscillations are again limited.

In our work, we use the second method and restrict ourselves to a finite dimensional subspace of curves, thereby using a sparse representation. Thus the oscillatory nature of deformations of a curve depends on the basis of the finite dimensional space of curves, and cannot be greater then the oscillations in the basis. We can use any finite dimensional representation for a curve, for example splines [69, 82] or fourier descriptors [88]. Representing and manipulating shapes based on splines can be seen in [63] [16], while similar work using fourier descriptors can be found in [77, 14].

4.3 Evolving curves by solving an Ordinary differential equation in a subspace

In order to restrict curve evolution to the desired finite dimensional subspace, we look at an analogous and in some sense a more general problem, that of evolving a point in a space according to a defined velocity field. Let X be an infinite dimensional space with an inner product $\langle -, - \rangle_X$ and E be its finite dimensional linear subspace. We can identify $\mathbb{R}^n$ and E (dim(E) = n) and use the standard inner product $\langle -, - \rangle_{\mathbb{R}^n}$. Let us define a curved subspace M (dim(M) = q) of E by some constraint implicit equations, $g(x) = 0, \forall x \in M$, where $g : \mathbb{R}^n \to \mathbb{R}^m$ is a submersion so that we have a tangent space defined at all points of M, i.e. $M = g^{-1}(0)$.

We are interested on trajectories constrained to stay in M, and given by the evolution equation

$$\frac{dx(t)}{dt} = \vec{V}(x) \tag{4.1}$$

with $x(t) \in M, \forall t \geq 0$. This requirement implies that the velocity $\vec{V(x)}$ is tangent to M at $x(t)$. But in general, it is easy to see that this may not be true. For instance many evolutions are obtained as the gradient descent flows from a cost-function $\mathcal{L} : E \to \mathbb{R}$, the steepest descent flow having the form

$$\frac{dx}{dt} = -\nabla_E \mathcal{L}_{x(t)}$$

and the gradient $\nabla_E \mathcal{L}_{x(t)} \in T_{x(t)}E$ does not need to belong to $T_{C(t)}M$. Some projection steps are thus needed. The situation becomes even more problematic when the equation is discretized:

$$x(t + dt) = x(t) + dt\vec{V}(x).$$

If M is linear, it is enough to project $\vec{V}$ orthogonally on M to ensure that the constraints are satisfied, but there is no guarantee that $x(t + dt)$ lies in M when M is curved. Via the projection $P^M_{x(t)}(\vec{V}(x))$ of the velocity on $T_{x(t)}M$, the exponential map will provide an evolution:

$$x(t + dt) = \exp^M_{x(t)} \left(dt P^M_{x(t)}(\vec{V}(x)) \right)$$

(see [13] for an example). This map may turn to be complicated to compute, specially in our case where we are given an implicit definition of the subspace. So we propose to approximate the exponential map via a "descent on the level sets of g". With this notation, our scheme is written as

$$x(t + dt) = P^g_{x(t)} \left(dt P^M_{x(t)}(\vec{V}(x)) \right). \tag{4.2}$$

where $P^g_{x(t)}$ can be seen as an approximation of the exponential map and is a descent along the level sets of the (square) Euclidean norm of the constraint function g which defines M. The first step consists in projecting the velocity orthogonally in $T_{x(t)}M$. That will turn to be easy. Set $\tilde{x}(t + dt) := x(t) + dtP^M_{x(t)}(\vec{V}(x))$. Then, in general, $\tilde{x}(t + dt) \notin M$ and $g(\tilde{x}(t + dt)) = a \neq 0$. Then we find $x(t + dt)$ by steepest descent on $\|g\|^2$, starting at $\tilde{x}(t + dt)$ in order to reach the level set $g = 0$.

Next we discuss the heirarchy of the space of curves we consider for the curve evolution problem before going on to explain the projection steps in detail.

4.4 Space of Curves

Given a smooth manifold X and $x \in X$, T_xX will denote in the sequel the tangent space of X in x. When X is a submanifold of the Riemannian manifold Y, then $T_xX \subset T_xY$ and there is an orthogonal projection $T_xY \to T_xX$ (see [6] for instance), we will denote it by P^X_x.

As a general manifold of curves, one considers generally the set of $\mathcal{C}^p$ 1-dimensional manifolds of $\mathbb{R}^2$, $B_e(\mathbb{S}^1, \mathbb{R}) = \text{Emb}^p(\mathbb{S}^1, \mathbb{R}^2)/\text{Diff}^p(\mathbb{S}^1)$, the space of all embeddings of the unit circle $\mathbb{S}^1$ into the plane $\mathbb{R}^2$ with curves having same geometrical image being identified. It is equipped with the following H^0 Riemannian metric: given $c : [0, L] \to \mathbb{R}^2$ a parameterized curve in B_e, the inner product on $T_c B_e$ is given by

$$\langle h, k \rangle_{B_e} = \int_0^L h(p)k(p)|c'(p)|dp \tag{4.3}$$

where $h(p)$ and $k(p)$ are parameterized normal deformations of c, i.e. smooth normal vector fields defined along c (see [51]). We call this inner product the geometric inner product and it can be induced onto any subspace of it.

Let $\mathcal{E}$ denote the space of $\mathcal{C}^p$ embeddings $\text{Emb}^p(\mathbb{S}^1, \mathbb{R}^2)$. We will not work on $\mathcal{E}$, but on a finite dimensional submanifold of it, we will assume the following settings: we are given a finite dimensional subspace V of $\mathcal{C}^p(\mathbb{S}^1, \mathbb{R}^2)$, and set $E = V \cap \mathcal{E}$. The rational behind this construct is to consider a linear isomorphism. i.e a parametrization of regular maps $\mathbb{S}^1 \to \mathbb{R}^2$:

$$\pi : \mathbb{R}^m \to V$$

whose image will consist of embeddings when out of a critical set of parameters configurations. Specific constraints are applied by restricting it further to a set $M \subset g^{-1}(0) \cap \mathcal{E}$ where $g : V \to \mathbb{R}^n$ is a subimmersion (i.e. the tangent map of g at this point has maximal rank), so that, at least locally, M is a submanifold of E

$$M \subset E \subset \mathcal{E}.$$

Another valid inner product on the space V is given by considering the isomorphism to $\mathbb{R}^m$ and lifting the inner product on $\mathbb{R}^m$.

$$\langle h, k \rangle_{\mathbb{R}} = \pi^{-1}(h)^t \pi^{-1}(k) \tag{4.4}$$

where $\pi^{-1}(h), \pi^{-1}(k)$ are the finite dimensional representation of the deformation fields and $\pi^{-1}(h)^t$ denotes transpose of $\pi^{-1}(h)$. A conceptual diagram depicting the hierarchy of the spaces is given below. Thus the space $\mathcal{E}$ can be induced with an inner product from the space B_e, the geometric inner product (4.3), while the linear space E and its curved subspace M have two inner products defined on them, the geometric inner product and the $\mathbb{R}^m$ inner product (4.4).

4.5 A series of Projection steps

4.5.1 The projection P_C^V

We first project the given velocity vector $v \in T_c\mathcal{E}$ or $v \in T_c B_e$ to $T_c V$, the tangent space to the linear subspace V. Let $e_1, \ldots, e_m$ be the basis of $T_c V$. For a given vector $v \in T_c\mathcal{E}$, we compute its orthogonal

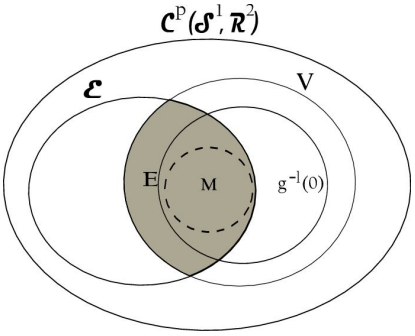


Figure 4.1: A conceptual diagram showing the hierarchy of the spaces that we work with

projection $P_C^V(v) = \sum_{i=1}^m v_i^V\ e_i$ by requiring that $v - P_C^V(v)$ must be orthogonal to T_cV:

$$\langle v - P_C^V(v), e_i \rangle = 0, \quad i = 1, \ldots, m$$

which gives the system of equations on the v_i^V

$$\sum_{j=1}^{m} \langle e_i, e_j \rangle v_j^V = \langle v, e_i \rangle, \quad i = 1, \ldots, m. \tag{4.5}$$

The matrix $G_1 = (\langle e_i, e_j \rangle)_{ij}$ is known as the Gram matrix of the family e_i (see [49]) and is given as

$$G_1 = (\langle e_i, e_j \rangle_{B_e})_{ij} = \int_0^L \langle e_i(p), \mathbf{n}(p) \rangle_{\mathbb{R}^2} \langle e_j(p), \mathbf{n}(p)) \rangle_{\mathbb{R}^2} |c'(p)|\, dp$$

while the i-component of the second member is given by

$$\langle v, e_i \rangle_{B_e} = \int_0^L \langle v(p), \mathbf{n}(p) \rangle_{\mathbb{R}^2} \langle e_i(p), \mathbf{n}(p)) \rangle_{\mathbb{R}^2} |c'(p)|\, dp.$$

where $\mathbf{n}(p))$ is the inner unit normal to the curve, $|c'(p)|$ is the speed of the curve c and $\langle -, - \rangle_{\mathbb{R}^2}$ is the usual scalar product in $\mathbb{R}^2$.
The orthogonal projection we aim at is $P_C^V(v) = G_1^{-1}(\langle v, e_i \rangle_i)$.

4.5.2 The map P_C^M

Using the Gram matrix method, we need to project $v = \sum_i^m v_i^V e_i \in T_cV$ to T_cM. Let us denote the vector of co-efficients of v as v^V. From its definition as the subset of $M = g^{-1}(0)$ of the points $x \in M$ where T_xg has maximal rank, T_cM is the null space (or kernel) of the differential $dg(c)$. Let $m_{c1}, \ldots, m_{c\ell}$ is a basis of T_CM(the subscript c explicitly reminds us that the basis vary with the point c). We compute the orthogonal projection $P_C^M(v) = \sum_{i=1}^{\ell} v_i^M\ m_{ci}$ by requiring that $v - P_C^M(v)$ must

be orthogonal to $T_C M$:

$$\langle v - P_C^M(v), m_{ci} \rangle = 0, \quad i = 1, \ldots, \ell$$

which gives the system of equations on the v_i^M

$$\sum_{j=1}^{m} \langle m_{ci}, m_{cj} \rangle v_j^M = \langle v, m_{ci} \rangle, \quad i = 1, \ldots, \ell. \tag{4.6}$$

The matrix $G_2 = (\langle m_{ci}, m_{cj} \rangle)_{ij}$ is known as the Gram matrix of the family m_{ci} (see [49]). Depending on the inner product, it is given as,

$$G_2 = (\langle m_{ci}, m_{cj} \rangle_{\mathbb{R}})_{ij} = (\pi^{-1}(m_{ci}))^t \, \pi^{-1}(m_{cj})$$

or

$$G_2 = (\langle m_{ci}, m_{cj} \rangle_{B_e})_{ij} = \int_0^L \langle m_{ci}(p), \mathbf{n}(p) \rangle_{\mathbb{R}^2} \, \langle m_{cj}(p), \mathbf{n}(p) \rangle_{\mathbb{R}^2} |c'(p)| \, dp$$

while the i-component of the second member is given by

$$\langle v, m_{ci} \rangle_{\mathbb{R}} = (v^V)^t \, \pi^{-1}(m_{ci})$$

or

$$\langle v, b_i \rangle_{B_e} = \int_0^L \langle v(p), \mathbf{n}(p) \rangle_{\mathbb{R}^2} \langle m_{ci}(p), \mathbf{n}(p) \rangle_{\mathbb{R}^2} |c'(p)| \, dp.$$

The first set uses the $\mathbb{R}^m$ inner product and the second set uses the geometric inner product on curves. The orthogonal projection we aim at is $P_C^M(v) = G_2^{-1}(\langle v, m_{ci} \rangle_i)$. Finally we have to project the vector on $T_C M$ to M.

4.5.3 The map P_C^g

After $P_{C(t)}^M(v)$ has been computed, we need to map this increment into a destination $c(t+dt)$ on M. We proceed as follows. Define $f : E \to \mathbb{R}$ by $f(x) = \|g(x)\|^2$ so that if $x \in M$, $f(x) = 0$.

Set $\tilde{c}(t+dt) = c(t) + dt P_{C(t)}^M(v)$. We define a curve family $D(-, \tau)$ with $D(-, 0) = \tilde{c}(t+dt)$ and we evolve it under the differential equation

$$\frac{\partial D}{\partial \tau} = -\nabla_E f(D(\tau))$$

until steady-state and define

$$P_{C(t)}^g \left(dt P_{C(t)}^M(v) \right) := D(\infty).$$

The solution of this flow can be implemented using a standard scheme. e.g:

$$D(\tau + d\tau) = D(\tau) - d\tau \nabla_E f(D(\tau)).$$

Although started in E, there is no guarantee that this stays in E or even if $D(\tau)$ stays in E for all τ's, the steady state might not be in E. It will nevertheless stay in V since V is *linear*, finite dimensional, hence closed. This implies that singular configurations may be reached. We did in fact encounter the problem in one of the experiments presented in Section 4.9 where an evolving curve started to develop self intersection, thus failing to be embedded.

4.6 Working with Coordinates

The manipulation of the different projections is normally performed in a numerical space of parameters for E, say $\mathbb{R}^m$ if $\dim V = m$ via a linear isomorphism $\pi : \mathbb{R}^m \to V$ by choosing a basis $(e_1, \ldots, e_m)$ of V, where we may assume that each e_i is a curve parametrized as $e_i : \mathbb{R} \to \mathbb{R}^2$, of class $\mathcal{C}^p$ and periodic of period say L:

$$\pi : (\lambda_1, \ldots, \lambda_m) \mapsto s \in [0, L] \mapsto \sum_{i=1}^{m} \lambda_i e_i(s).$$

Under this isomorphism, E is realized as the subset $\bar{E} := \pi^{-1}(E)$. Set $\bar{g} = g \circ \pi$.

$$\bar{g} : \mathbb{R}^m \to \mathbb{R}^n, \quad x \mapsto (\bar{g}_1(x), \ldots, \bar{g}_n(x)).$$

Then $\bar{M} := \pi^{-1}(M) = \bar{g}^{-1}(0)$ where $\bar{g}$ is a subimmersion, which means that the Jacobian $J\bar{g}$ of $\bar{g}$ has maximal rank.

Let $(\epsilon_1, \ldots, \epsilon_m)$ the natural basis of $\mathbb{R}^m$. Then we can compute the *metric induced from* E on $\mathbb{R}^m$ as

$$\langle \epsilon_i, \epsilon_j \rangle_{\pi, B_e} = \int_0^L \langle e_i(p), \mathbf{n}(p) \rangle_{\mathbb{R}^2} \langle e_j(p), \mathbf{n}(p) \rangle_{\mathbb{R}^2} |C'(p)| \, dp$$

up to a multiplicative constant depending on the period. The scalar product is totally described by the Gram matrix $G = (\langle e_i, e_j \rangle)_{i,j}$. In order to project orthogonally on $T_x\bar{M}$ with respect to the above metric of $\mathbb{R}^m$ restricted to $T_x\bar{M}$, we need first a basis of $T_x\bar{M}$ that allows to carry the projection construct from the previous paragraph and such a basis can be extracted by standard procedure from the Jacobian $J\bar{g}$ at x. Having such a basis $m_{x1}, \ldots, m_{x\ell}$ allows to compute the projection $P_x^{\bar{M}}$ corresponding to $P_{\pi(x)}^{M}$.

If $\bar{f}$ is the expression of $f : E \to \mathbb{R}$ in the coordinates provided by π, let $\nabla_G \bar{f}$ be the gradient of $\bar{f}$ for the metric (G) induced by E. There is in fact a simple relation between this gradient and the usual gradient on $\mathbb{R}^m$, denoted $\nabla \bar{f}$:

$$\langle \nabla_G \bar{f}, x \rangle_G = \nabla_G \bar{f}^T G \, x = \left(G^T \nabla_G \bar{f} \right)^T x = \langle G^T \nabla_G \bar{f}, x \rangle_{\mathbb{R}^m}$$

where the last scalar product is the usual product on $\mathbb{R}^m$ and therefore $G^T \nabla_G \bar{f}$ is the gradient $\nabla \bar{f}$ of $\bar{f}$ for the usual scalar product of $\mathbb{R}^m$ and since G is symmetric, it follows that

$$\nabla_G \bar{f} = G^{-1} \nabla \bar{f} \tag{4.7}$$

and the partial derivative $\bar{f}_{x_j}$ are given by

$$\bar{f}_{x_j}(x) = 2\sum_{i=1}^{n} \bar{g}_i(x)\bar{g}_{x_j}(x). \tag{4.8}$$

G consists of inner products between the basis, either geometric or the usual euclidean inner product($G = Id$). Note that the Gram matrix using geometric inner product changes at every point. An important point to underline is the fact that, although the gradient is a metric dependent entity, a critical point, i.e. a point x such that $\nabla f(x) = 0$ is not, since it is in fact a point where $T_x f = 0$ and the tangent map is not metric dependent. It raises the question on whether we can use the ambient metric of $\mathbb{R}^m$ instead of the geometric one. Because $f^{-1}(0)$ is not reduced to a single point, changing the gradient may indeed change the projection P^g since it will generally change the trajectory that reaches $f^{-1}(0)$. Some experiments illustrating this point will be presented in Section 4.9.

The evolution scheme in $\mathbb{R}^m$ is summarized more algorithmically in Figure 4.2. Depending on the

1. Let $c = c(-, 0)$ the initial curve in $\mathcal{E}$. Set $x(0) = \pi^{-1}(c)$
2. Loop until a stopping criterion is fulfilled:
   (a) compute the velocity $\vec{V}(t)$ associated to $c(-, t) \in \mathcal{E}$ and set
   $$v(t) = G_1^{-1}[\langle e_i(s), \vec{V}(s, t)\rangle]_{i=1\ldots m}$$
   the image in $\mathbb{R}^m$ of the orthogonal projection on $V(m$ dimensional linear subspace) of $\vec{V}(t)$.
   (b) compute a basis of $T_{c(t)}\bar{M}$ (denote as m_{ci}) through $\bar{g}$.
   (c) compute the orthogonal projection $w(t)$ defined by G_2 of $v(t)$ in $T_{c(t)}\bar{M}$ as,
   $$w(t) = G_2^{-1}[\langle v(t), m_{ci}\rangle]_{i=1,\ldots,\ell}$$
   (d) form the vector $c(t, 0) = c(t) + dtw(t)$,
   (e) compute numerically the steady state of the flow
   $$\frac{\partial c(t, \tau)}{\partial \tau} = -\nabla_G \bar{f}(c(t, \tau))$$
   the gradient $\nabla_G \bar{f}$ being computed via equations (4.7),(4.8), The Gram matrix G is either computed using the geometric inner product, or the Euclidean inner product ($G = Id$).
   (f) set $c(t + dt) = c(t, \infty)$.
3. output the final result $c(\bar{t})$.

choice of inner products, the Gram matrices and other inner products are given in the following table.

In the next section we apply it to a non trivial example of spline curves where an equidistance condi-

1. Geometric Inner Product:

$$G_1 = (\langle e_i, e_j \rangle_{B_e})_{ij} = \int_0^L \langle e_i(p), \mathbf{n}(\mathbf{p}) \rangle_{\mathbb{R}^2} \langle e_j(p), \mathbf{n}(\mathbf{p}) \rangle_{\mathbb{R}^2} |C'(p)| \, dp$$

$$\langle \vec{V}, e_i \rangle_{B_e} = \int_0^L \langle \vec{V}(p), \mathbf{n}(\mathbf{p}) \rangle_{\mathbb{R}^2} \langle e_i(p), \mathbf{n}(\mathbf{p}) \rangle_{\mathbb{R}^2} |C'(p)| \, dp.$$

$$G_2 = (\langle m_{Ci}, m_{Cj} \rangle_{B_e})_{ij} = \int_0^L \langle m_{Ci}(p), \mathbf{n}(\mathbf{p}) \rangle_{\mathbb{R}^2} \langle m_{Cj}(p), \mathbf{n}(\mathbf{p}) \rangle_{\mathbb{R}^2} |C'(p)| \, dp$$

$$\langle v, b_i \rangle_{B_e} = \int_0^L \langle v(p), \mathbf{n}(\mathbf{p}) \rangle_{\mathbb{R}^2} \langle b_i(p), \mathbf{n}(\mathbf{p}) \rangle_{\mathbb{R}^2} |C'(p)| \, dp.$$

2. $\mathbb{R}^m$ inner product:

$$G_2 = (\langle m_{Ci}, m_{Cj} \rangle_{\mathbb{R}})_{ij} = (\pi^{-1}(m_{Ci}))^t \, \pi^{-1}(m_{Cj})$$
$$\langle v, m_{Ci} \rangle_{\mathbb{R}} = v_c^t \, \pi^{-1}(m_{Ci})$$

tion is imposed in the node points.

4.7 An Example: Euclidean Shortening Flow for Splines with $\mathbb{R}^2$-Equidistant Node Points

We apply the previous theory to the case where E is the space of order n (rational uniform) B-splines closed curves with N nodes. We impose the constraint that three consecutive nodes P_r, P_{r+1}, P_{r+2} are equidistant in $\mathbb{R}^2$ (and not with respect to arc-length), i.e. $\|P_{r+2} - P_{r+1}\| = \|P_{r+1} - P_r\|$, this provides the subspace M of E. We consider the descent flow arising from the length functional of a closed curve C parametrized by $c : [0, L] \to \mathbb{R}^2$,

$$\mathcal{L}(C) = \int_0^L |c'(p)| dp$$

the associated flow being the classical Euclidean shortening flow

$$\frac{\partial c}{\partial t} = \kappa \mathbf{n}$$

where $\kappa \mathbf{n}$ is the curvature vector of C. We want to restrict it to the space of regular spline curves with equidistant node points, as it is well known that not imposing conditions on the node points of the spline curves lead generally to instability under this flow.

Let V the space of closed spline curves of order n and passing through N given node points of $\mathbb{R}^N$. It is well known that the choice of these N points, when N is large enough with respect to n, defines a

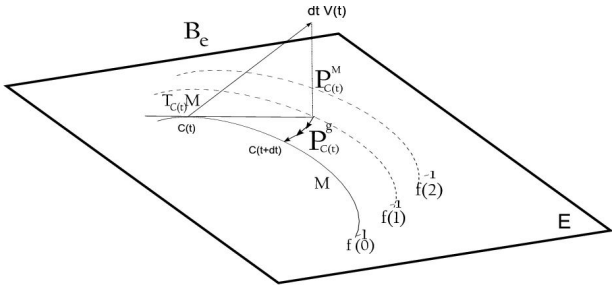


Figure 4.2: An evolution step in M

linear isomorphism

$$\pi : (\mathbb{R}^2)^N \to V.$$

Details can be found for instance in [83, 3, 7]. Some node configuration will lead to spline curves that fail to be embedding, because of self intersection or regression points for instance, so in order to have embedding we need to restrict the configurations to an open set of $(\mathbb{R}^2)^N$. $(\mathbb{R}^2)^N$ is equipped with the natural basis formed by the vectors $e_{1x}, e_{1y}, \ldots, e_{Nx}, e_{Ny}$ where e_{ix} (resp e_{iy}) is the vector $(1, 0)$ (resp $(0, 1)$) on the i-th factor of $(\mathbb{R}^2)^N$.

The equidistance constraint is most naturally defined on the coordinate space $(\mathbb{R}^2)^N$ and is given by

$$\begin{aligned} \bar{g} : (\mathbb{R}^2)^N &\to \mathbb{R}^{N-1}, \\ (P_0, \ldots, P_{N-1}) &\mapsto (d_{21} - d_{10}, d_{32} - d_{21}, \ldots, d_{N-1\,0} - d_{N-1\,N-2})^T \end{aligned}$$

where $d_{ij} := \|P_j - P_i\|^2$ is the squared Euclidean distance between P_i and P_j. The zero-level set of $\bar{g}$ represent "closed" configurations of N points in $\mathbb{R}^2$ such that the two segments that link 3 consecutive of these points have the same length. In an imaged way, we may call such a configuration a N-links bicycle chain. The constrained manifold M_N in which we carry the projected Euclidean shortening flow will consist of the regular spline curves generated by such an N-links bicycle chain. Note that $g^{-1}(0)$ is the intersection of quadratic hypersurfaces. Its geometry is very rich and tools from algebraic geometry are needed to describe it more thoroughly, but this is out of the scope of this paper.

4.8 An Application: Restricted Geodesic Active Contours

Geodesic active contours is a method to detect object boundary contour. A user initializes a contour and the contour then evolves to adhere to the object boundary. The curve evolves with a velocity that

takes it closer to the object boundary. Hence the only change that we need to make in the previous implementations, is to change the velocity vector that we need to project to the N-links bicycle chain subspace. As we show in the next section, the unrestricted active contour may cause the node points to collide with each other causing the evolution to collapse. Restricting the contours to the m-links chain subspace, adds a tangential motion to the points, causing them to stay equidistant. The velocity vector is given as,

$$v = \frac{\partial c}{\partial t} = (\kappa g - \langle \nabla g, \mathbf{n} \rangle + \alpha g)\mathbf{n} \tag{4.9}$$

where $\mathbf{n}$ is the inner unit normal to the curve and g is a function of the image gradient at the curve points, $g : [0, +\infty[\rightarrow [0, +\infty[$ which is a regular monotonically decreasing function. α is a positive constant which makes detection of non convex objects easier and speeds up the convergence. For details of the geodesic active contour algorithm, we refer the reader to [1].

4.9 Experiments

We now run some experiments for restricting the curvature flow to the subspace M_N. We are mainly interested in observing what properties of the original curvature flow are preserved by the projected one. We start with a very simple configuration: we evolve a four point curve, shown in Figure 4.9. We display the evolution of the spline curve as well as the evolution of the rhombus formed by the node points. We can see that a rhombus evolves into a limiting square. We state a conjecture that, if we join the N node points of a simple spline curve and run curvature flow on it, then the polygon should evolve into a regular $N-gon$. This would provide a result analogous to the one proved by Grayson in [31], that the curvature flow shrinks simple planar curves to round points.

Next we take a comb-like curve with 30 node points and evolve it with the curvature flow in Figures 4.4,4.5. This curve has portions that are varying slowly and some portions that are wildly oscillating. We observe that both theevolutions converge to a round point, but they follow different paths.

We annotate a human hand with 30 points and evolve it via the curvature flow in Figures 4.6,4.7. In both examples, we see that self-intersections can occur in any of the evolution. We observe that in the evolution using geometric inner product, the oscillations disappear the fastest (in the sense that the first convex curve obtained in the evolution has a larger area).

Finally we implement the modified geodesic active contours. We restrict all the intermediate contours to the subspace of m-links chain ($m = 30$, in this case). We see that the node points collide and then onwards the evolution collapses, as shown in Figure 4.8, if the nonlinear constraint is not enforced. There could be problems with self-intersection in the case where the constraint is enforced, as shown in one of the evolutions that uses the Euclidean inner product, refer Figures 4.9,4.10. Although one can still continue the evolution, certainly it is not desirable to have self-intersections. This happens clearly due to the fact that the constraints do not explicitly require a curve not to have self-intersections.

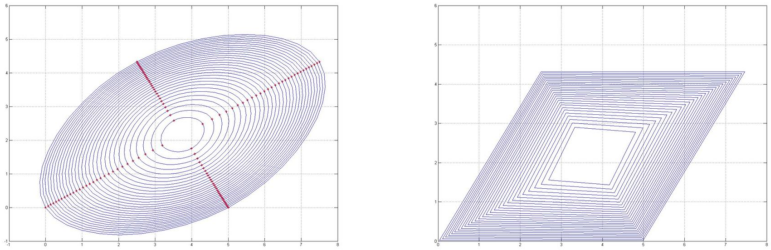


Figure 4.3: (left) Curvature flow of a rhombus restricted to M_4, points shown in red '*' and 'o' are the node points. (right) Node points of curves in M_4 during curvature flow joined by line segments

4.10 Conclusion

We have described a numerical scheme for implementing a flow of parameterized curves in a non linear subspace of a finite dimensional subspace of embeddings, using a "descent across level sets" to approximate the exponential map. We have applied it to the Euclidean shortening flow for spline curves with equidistant node points and demonstrated the soundness of this approach through several experiments. We have an ongoing research on non linear constraints that restrict the sets of admissible shapes, and we are developing a full geometrical framework for the evolution of 1-dimensional regular manifolds of $\mathbb{R}^2$, by ensuring parametrization independence. The formalism is very similar but leads to somewhat more complex projection steps, a trade-off that must be considered in applications.

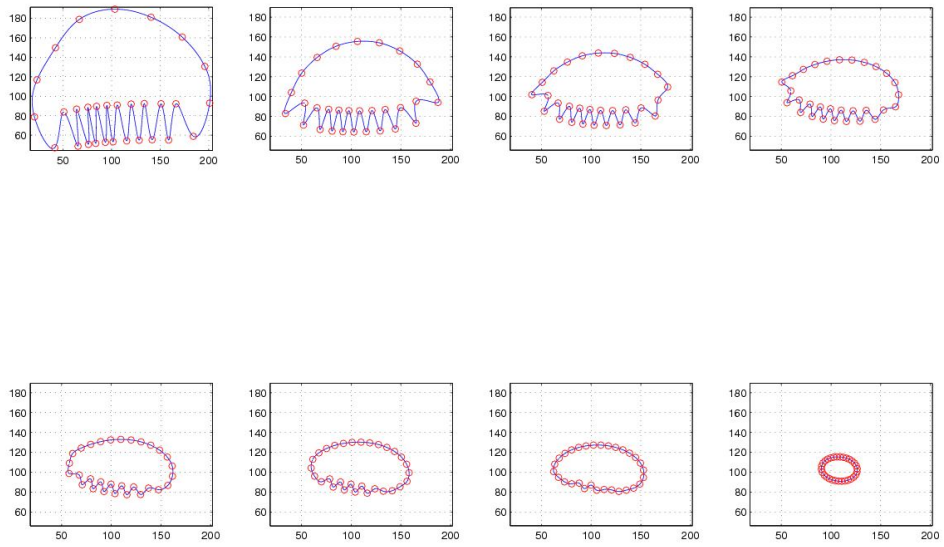


Figure 4.4: Some iterations showing curvature flow of a comb like curve, using the $\mathbb{R}^{2N}$ inner product

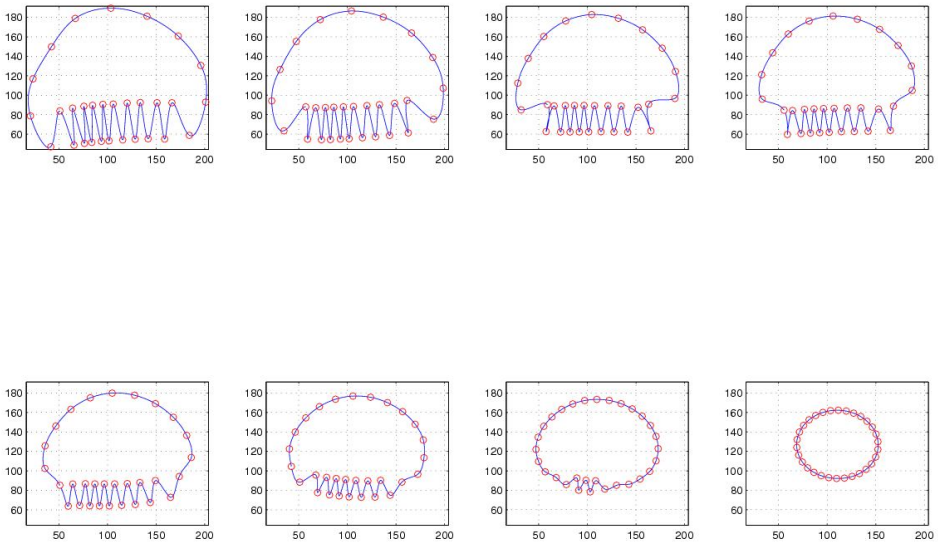


Figure 4.5: Some iterations showing curvature flow of a comb like curve, using the geometric inner product

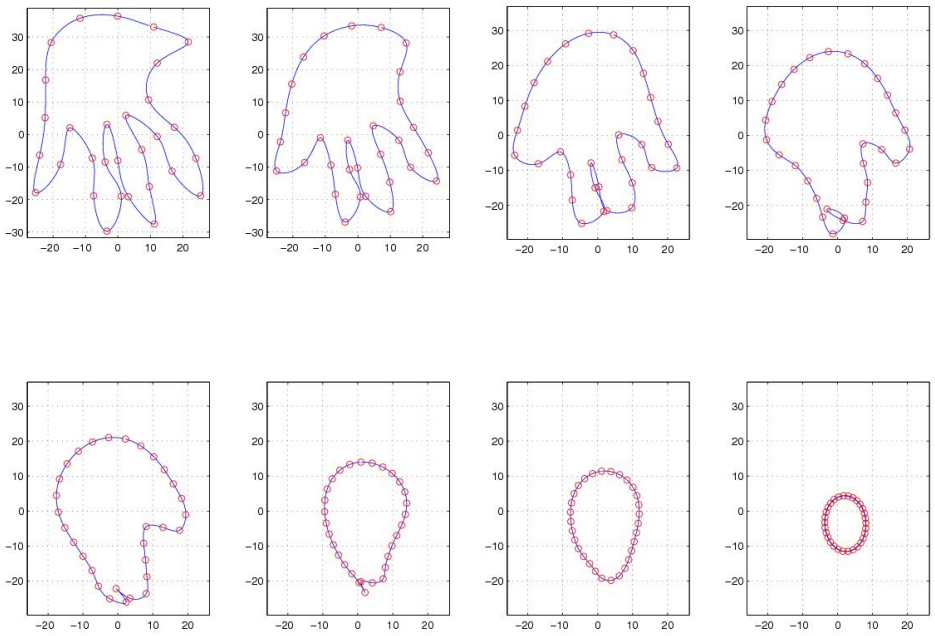


Figure 4.6: Some iterations showing curvature flow of a 'human hand' like curve, using the $\mathbb{R}^{2N}$ inner product

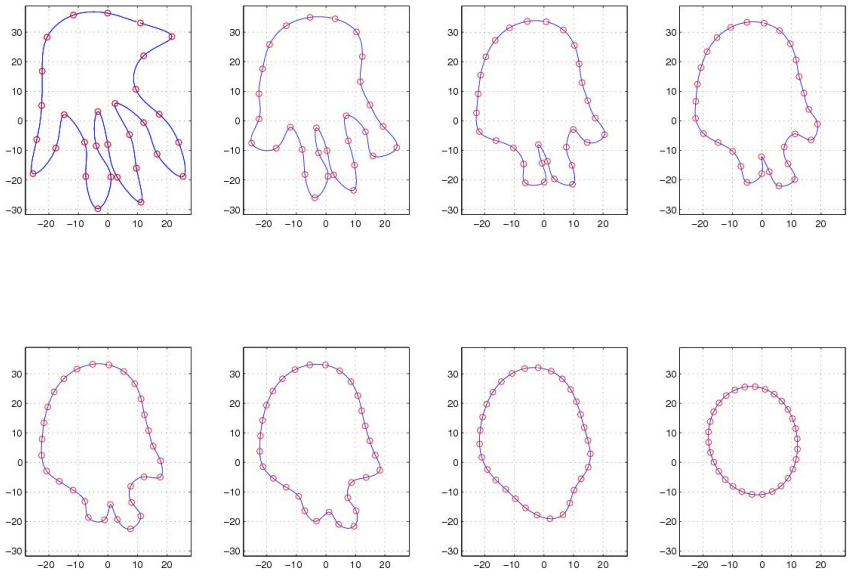


Figure 4.7: Some iterations showing curvature flow of a 'human hand' like curve, using the geometric inner product

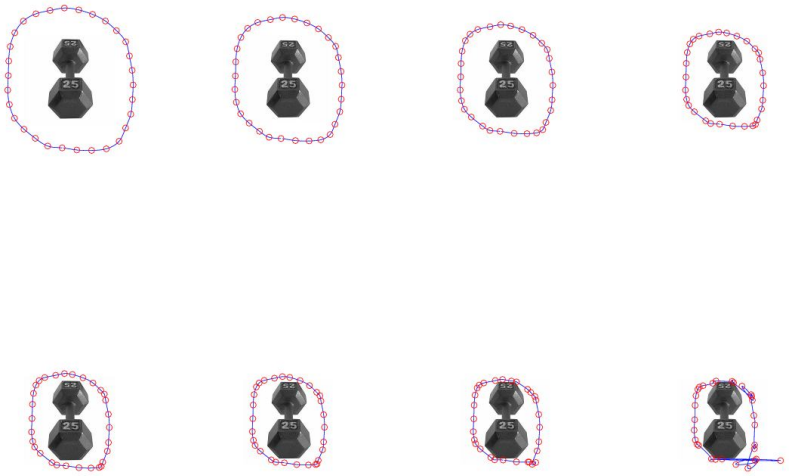


Figure 4.8: Some iterations showing the unrestricted geodesic active contours

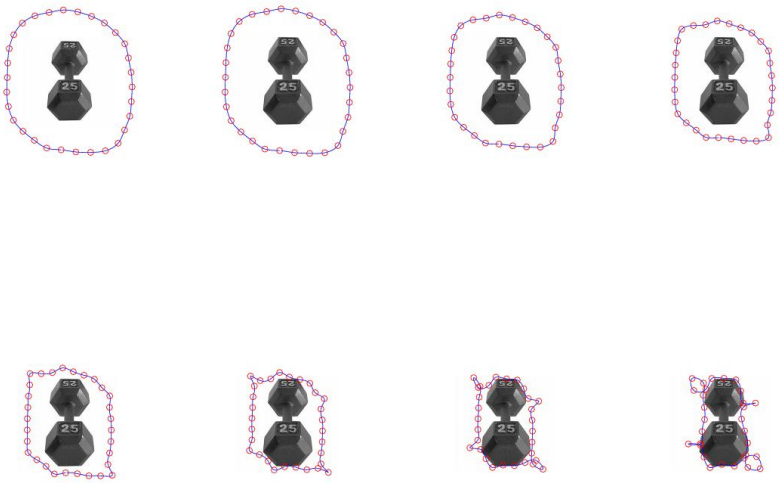


Figure 4.9: Some iterations showing the restricted geodesic active contours using the $\mathbb{R}^{2N}$ inner product

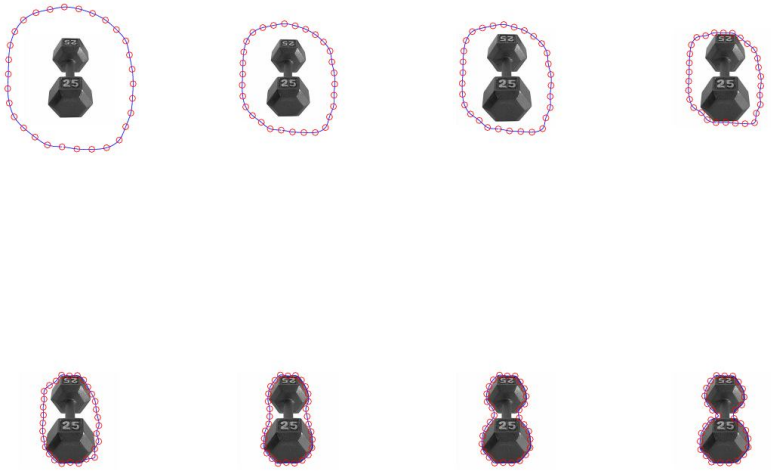


Figure 4.10: Some iterations showing the restricted geodesic active contours using the geometric inner product

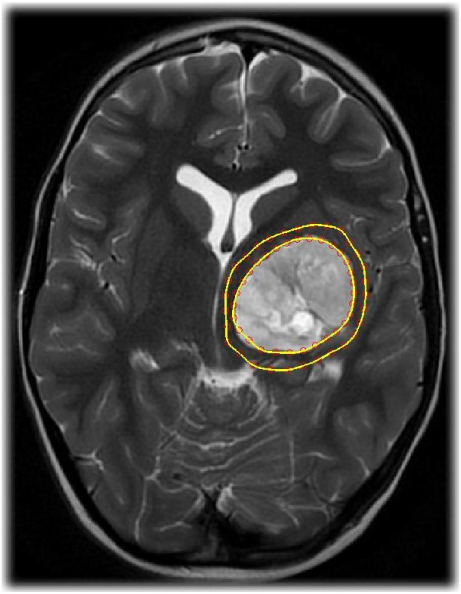


Figure 4.11: Brain tumor geodesic active contour segmentation using the geometric inner product

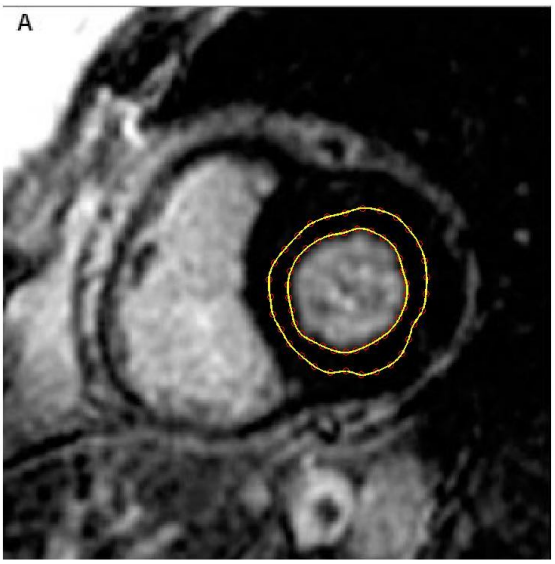


Figure 4.12: Heart MRI image segmentation using the geodesic active contours with the geometric inner product

Chapter 5

On Restricting Curve Evolution to Finite Dimensional Implicit Subspaces with Non-Euclidean Metric

We plan to submit this chapter as *On Restricting Curve Evolution to Finite Dimensional Implicit Subspaces with Non-Euclidean Metric* to *Journal of Mathematical Imaging and Vision*.

5.1 Introduction

Curve evolution has now become a standard tool in Computer Vision. It has been used for tracking interfaces, registration, active contour algorithms for segmentation, etc. [38, 72, 44, 3]. Curve evolution involves deformation or motion of a curve via a given velocity vector field defined on a space of curves that share some common structures. Curves can be manipulated implicitly as the zero-level set of a given function for instance, or explicitly via a parametrization. In this paper we are interested in the latter. We work on the space of smooth parameterized curves, which is a Riemannian manifold.

The general curve evolution equation is given as,

$$\frac{\partial c}{\partial t}(p,t) = v(p,t)\vec{n}(p,t) + w(p,t)\vec{t}(p,t) \tag{5.1}$$

where $c(-,-)$ is the family of curves satisfying the equation, p is the parameter along a curve, t is the time parameter, v is the scalar velocity in the unit normal direction $\vec{n}$ of the curve, w is the scalar velocity in the tangential direction $\vec{t}$ to the curve. In [23], the authors show that the tangential component of the velocity affects only the parameterization of the curve, thus can be omitted and we get

$$\frac{\partial c}{\partial t}(p,t) = v(p,t)\vec{n}(p,t) \tag{5.2}$$

Curve evolution equations arise either as a 'direct design' approach where the velocity vector is given according to some application specific requirement, or as a gradient descent scheme in order to minimize an energy functional $E : \mathbb{S} \to \mathbb{R}$, where $\mathbb{S}$ is the space of curves in which we are interested. A fact that has been overlooked until recently is that the gradient of the functional depends on the definition of the inner product on $\mathbb{S}$. One of the most favourite inner products has been the standard reparameterization invariant L_2 inner product between *normal* curve deformations v_1 and v_2 of a curve $c(p)$:

$$\langle v_1, v_2 \rangle_c = \int_0^1 v_1(p) v_2(p) |c'(p)| dp \tag{5.3}$$

In what follows, we denote this inner product as $\langle -, - \rangle_c$, where c is the point, i.e. a curve, at which it is computed. When there is no ambiguity, we may omit the subscript c. In [12], the authors emphasize the importance of choosing the proper inner product for minimizing various energies. In [78], the authors propose to use Sobolev inner products for active contour segmentation, thereby avoiding local minima amongst other advantages.

Curve evolution is also implicitly used in applications like shape matching and classification based on shapes, since it involves deforming or evolving one shape into another. The space of curves $\mathbb{S}$ under consideration may vary from application to application. The space of curves may not be a simple euclidean space(linear, flat metric), which makes it an ideal ground for use of differential geometric tools. The space of smooth embeddings $Emb(\mathbb{S}^1, \mathbb{R}^2)$ and smooth immersions $Imm(\mathbb{S}^1, \mathbb{R}^2)$, identifying reparameterizations gives rise to two spaces of curves:

$$B_e = Emb(\mathbb{S}^1, \mathbb{R}^2)/Diff(\mathbb{S}^1)$$

and

$$B_i = Imm(\mathbb{S}^1, \mathbb{R}^2)/Diff(\mathbb{S}^1)$$

(note $B_e \subset B_i$). In [51], the authors prove that the L^2 metric is a weak Riemannian metric: it degenerates on B_e and B_i, i.e. the geodesic distance between any two curves is zero. Given a curve, one can perturb it using infinitesimal deformation with very high frequencies towards any other given curve. These deformations are infinitely close to any curve and can be used to build arbitrarily short paths between any two curves, i.e. the corresponding geodesic distance degenerates. This implies that one cannot compute distance between shapes using this inner product, but segmentation algorithms would still work fine. There have been essentially two strategies to overcome this hurdle. One is to change the metric on the space of curves, for eg. in [78], the authors use a Sobolev metric and in [51] the authors use a curvature dependant metric, both of which restrict the spatial variation in the deformations. Glaunes et. al. [26] represent curves as a linear functional on vector fields over $\mathbb{R}^d$ and compare curves by proposing norms over the dual space of vector fields on $\mathbb{R}^d$. These norms depend on spatial derivatives, which limit the variation of the vector fields, prohibiting high frequency variations. In [55, 87], a group theoretic approach is used. The set of admissible deformations forms a group and then the problem to compute geodesic distance between two given curves is reduced to computing the geodesic distance

between two members of the group of deformations that map one curve into another. They also consider spatial derivatives on infinitesimal deformations for constructing the inner product, thereby limiting the spatial variations.

The other way is to work on a subspace of the infinte dimensional space of all curves, where the problem does not occur. The space of closed curves with finite Fourier expansion is used in [77] to represent curve deformations. There is an upper bound on the frequency of deformations that one can introduce in the finite dimensional case and hence the metric does not degenarate on this subspace. The finite dimensional case is important since even though theoretically the intended curve evolution happens on the infinte dimensional space of all curves, but when implemented on a computer, it is restricted to some finite dimensional subspace. In [80], the authors restrict curve evolution to linear finite dimensional subspace of curves.

In this paper, we use the latter approach, using a finite dimensional subspace of B_i. Due to application dependant specifications, this subspace will often be curved. We work on problems where such specifications are given implicitly.*. A gradient descent approach is used in [77] to project curve deformations from the tangent space of the given subspace of curves, to curves in the subspace, whereas in this work we build an exponential map to do this using optimal control theory. We investigate the cases where the metric on such finite dimensional spaces in non-Euclidean. The problem (of restricting curve evolution) is equivalent to solving an ordinary differential equation on a manifold.

$$\frac{dc}{dt}(-,t) = F(c(-,t))$$

with $c(-,t) \in \mathbb{S}$. In [34], the authors describe how to integrate ordinary differential equations on a finite dimensional submanifold of $\mathbb{R}^n$ using local co-ordinates such that the resulting solution preserves important properties. In our case, the finite dimensional subspace of the space of all curves is implicitly given, and is induced with a non-Euclidean inner product. In the context of deformable models, we show how to restrict curve evolution to the subspace of equidistant neighboring node points B-spline curve with N node points. In this particular example, the space of curves B_i induces the L_2 metric on the linear finite dimensional space of N node point closed spline curves and on its curved subspace given by the implicit constraint of equidistant neighbring node points of the spline curves.

This paper is organized as follows. In section 5.2 we state and describe the projected evolution problem and describe our algorithm from a general point of view, with particular emphasis on computation of Riemannian Exponential maps, Log maps and parallel transports. We also give a toy example demonstrating our method towards the end of the same section. In section 5.4, we describe a particular subspace of spline curves that we are interested in, and derive the necessary formulae. In section 6.4, we give some results of restricting the curve evolution to the subspace described in section 5.4. We implement the curvature flow and geodesic active contours and conclude in section 6.5. For the reader's convenience, we have collected basic facts from Riemannian geometry in Appendix A.

*Explicit constraints give a parameterization of the subspace, which can be used to compute standard differential geometric maps like the exponential map, and thus solve the problem

5.2 Projected Curve Evolution

Let us consider a finite dimensional linear subspace S^N of dimension N of the space of curves B_i. A set of implicit constraints $F : S^N \to \mathbb{R}^m$ defines an ℓ- dimensional subspace (non flat in general) of S^N, as $V = F^{-1}(0)$. Assuming here that the Jacobian $JF(x)$ of F is onto at every $x \in V$, i.e. that F is a submersion, ensures that V a smooth submanifold of S^N. We can identify S^N with $\mathbb{R}^N$ through an isomorphism, considering $\mathbb{R}^N$ to be the space of parameters representing curves in S^N. We have $V \subset S^N \subset B_i$. In section 5.4, we have a more complicated hierarchy with the space of spline curves, but we defer the discussion to that particular section.

Then, given an initial curve $c(-,0) \in V$ and a curve evolution equation as in Equation (5.2), we require a solution $c(-,t)$, such that $c(-,t) \in V$ for $t \geq 0$. Obviously the velocity vector may evolve a given curve out of V, hence we need some projection steps. In brief, one iteration of our algorithm consists of the following three steps:

1. Compute $v_S = \Pi_{T_{c(t)}S^N} vn$, the projection of velocity to the tangent space of S^N.
2. Compute $v_V = \Pi_{T_{c(t)}V} v_S$, the projection to the tangent space of the implicitly defined subspace V.
3. Compute $c(t+dt) = Exp_{c(t)}(v_V\ dt)$, the exponential map to map the velocity vector from the tangent space of V to the subspace V.

The orthogonal projections and the Exponential map above are with respect to the metrics on S^N and V, these metrics being the restriction of the L^2 metric (5.3) to these submanifolds. In other words, even though we restrict to curves that have finite dimensional representations, they should not be treated simply as points in $\mathbb{R}^N$ or a submanifold of $\mathbb{R}^N$. They are 1 dimensional submanifolds of $\mathbb{R}^2$ and so we should use an appropriate inner product. Which one? Since we are now dealing with *finite dimensional manifolds*, *we will not encounter the geodesic distance degeneracy proved in [51]* for the "standard" inner product defined in equation (5.3) as the corresponding geodesic distances on S^N and V must give back the topologies of these spaces (see [21]). The three projection steps are now described in details.

5.2.1 Projecting to the tangent space of S^N

We set $\vec{v} = v\vec{n}$ a (smooth) normal vector field along the curve $c \in S^N \subset B_i$ (thus an element of $T_c B_i$). Let $\{e_i\}_{i=1,\ldots,N}$ be a basis of $T_c S^N$. Then $v_S = \sum_{i=1}^N \alpha_i e_i$ is the projection of $\vec{v}$ if and only if $\vec{v} - v_S$ is orthogonal to $T_c S^N$ as a subspace of $T_c B_i$ i.e.

$$\langle \vec{v} - v_s, e_i \rangle = 0 \quad \forall i = 1, ..., N.$$

Set $\boldsymbol{\alpha}$ We set $G_1 := (\langle e_i, e_j \rangle_C)_{ij}$, this is the Gram matrix of $e = (e_1, \ldots, e_N)$. Then the projected velocity is given by

$$v_S = \Pi_{T_{c(t)}S^N}(vn) = \sum_i [G_1^{-1}(\langle v, e \rangle_C)]_i\ e_i \tag{5.4}$$

5.2.2 Projection to tangent space of V

From its definition as the set $V = F^{-1}(0)$, $T_C V$ is the null space (or kernel) of $DF(x)$, the differential of F. Let $m_{C1}, \ldots, m_{C\ell}$ is a basis of $T_C V$. We compute the orthogonal projection $v_V = \Pi_{T_{c(t)}V}(v_S) = \sum_{i=1}^{\ell} v_i \, m_{Ci}$ by requiring that $v_S - \Pi_{T_{c(t)}V}(v_S)$ must be orthogonal to $T_C V$:

$$\langle v_S - P_V(v_S), m_{Ci} \rangle_C = 0, \quad i = 1, \ldots, \ell$$

which gives a system of equations on $v_V = v_{i\,i=1,\ldots,\ell}$

$$\sum_{j=1}^{m} \langle m_{Ci}, m_{Cj} \rangle_C v_j = \langle v_S, m_{Ci} \rangle_C, \quad i = 1, \ldots, \ell. \tag{5.5}$$

The matrix $G_2 = (\langle m_{Ci}, m_{Cj} \rangle_C)_{ij}$ is known as the Gram matrix of the family m_{Ci} (see [49]) and is given as

$$G_2 = (\langle m_{Ci}, m_{Cj} \rangle_C)_{ij}$$

while the i-component of the second member is given by

$$\langle v_S, m_{Ci} \rangle_C$$

The orthogonal projection we aim at is

$$v_V = \Pi_{T_{c(t)}V}(v_S) = \sum_i [G_2^{-1}(\langle v_S, m_{Ci} \rangle_C)]_i \, m_{Ci} \tag{5.6}$$

5.2.3 Exponential map on the submanifold V

Since we discretize the curve evolution PDE:

$$c(t + dt) = c(t) + v_V dt$$

where dt is a finite time step, $c(t + dt) \notin V$ generally. We use the exponential map to project the tangent vector v_V to V, i.e.,

$$c(t + dt) = Exp_{c(t)}(v_V dt)$$

Computing the Exponential map on implicitly defined manifolds is not straightforward as in the case manifolds given via local parameterizations, where such a computation boils down to solving a classical second order ODE. We obtain instead the Exponential map via Optimal control theory [62, 36], along the lines of [19]. They use the Euclidean inner product on the implicitly defined submanifold V, whereas in our case, we have the L_2 inner product. The embedding space B_i and S^N have the L_2 inner product and we use the corresponding induced inner product on V. This is justified since, the points in V are not just points in $\mathbb{R}^N$, but they represent 1-D submanifolds of $\mathbb{R}^2$, i.e. curves.

Assume that our ℓ-dimensional submanifold $V = F^{-1}(0)$ embedded in S^N is now equipped with

some inner product G. We use the variational approach of minimizing the length of a path in a given direction that completely lies on the manifold V. Using optimal control theory [19], we define our Hamiltonian to be,

$$H(p,x,\mu) = -\frac{1}{2}\dot{x}^T G(x)\dot{x} + p^T\dot{x} + \sum_{i=1}^{m}\mu_i JF_i(x)\dot{x} \tag{5.7}$$

where p is an auxillary variable, $JF(x)$ is the Jacobian of F, μ is a Lagrange multiplier. The first term in the Hamiltonian is the cost function (square of length in this case) to be minimized and the last term is the constraint stating that the path should remain in V, i.e. $dF_i(x,\dot{x}) = \nabla F_i(x)^t\dot{x} = JF_i(x)\dot{x} = 0, \forall i$.

Pontryagin's maximum principle gives the necessary conditions for a maximum:

$$\frac{\partial H}{\partial \dot{x}} = 0 \tag{5.8}$$

$$\dot{p} = -\frac{\partial H}{\partial x} \tag{5.9}$$

Equation (5.8) in our case gives,

$$G(x)^{-1}p = \dot{x} - J_G F(x)^t\mu$$

where $J_G F(x)$ is the Jacobian of F with respect to the metric $G(x)$ and is given as $J_G F(x) = JF(x)G(x)^{-1}$ (i.e., it is a matrix containing transpose of the gradient vector of the individual functions F_i, with respect to the metric G). We know that $\dot{x} \in T_x V$ and $J_G F(x)^t\mu \in N_x V$, the normal space of V at x. Therefore, we have

$$\dot{x} = \Pi^G_{T_x V}(G(x)^{-1}p) \tag{5.10}$$

the orthogonal projection to the tangent space with respect to the inner product $G(x)$, and

$$\begin{aligned} -J_G F(x)^t\mu &= \Pi^G_{N_x V}(G(x)^{-1}p) \\ \mu &= -\left(J_G F(x)^t\right)^\dagger \Pi^G_{N_x V}(G(x)^{-1}p) \end{aligned} \tag{5.11}$$

where $(J_G F(x)^t)^\dagger$ is the pseudo-inverse of $J_G F(x)^t$ *for the metric* G ([**?**, 35]), we call it G-pseudo-inverse in the sequel, and we show how to compute it at the end of this section. From Equation (5.9), we get

$$\dot{p} = \frac{1}{2}\dot{x}^t DG_x(x)\dot{x} - \sum_{i=1}^{m}\mu_i HF_i(x)\dot{x} \tag{5.12}$$

where $DG_x(x)$ is the derivative of $G(x)$ with respect to x and $HF_i(x)$ is the Hessian of F_i at the point x. Now, given the initial conditions:

$$\begin{cases} \dot{x}(0) &= u \\ \mu(0) &= 0 \\ p(0) &= G(x(0))u \end{cases}$$

one computes the Exponential map using finite difference approximations of Equations (5.10), (5.11) and

(5.12).

The G-pseudo-inverse in Equation (5.11) is computed as follows. Let us denote $J_G F(x)^t$ by A. Since A^t is onto, A is injective. For an injective matrix A, the pseudoinverse is given as,

$$A^\dagger = (A^* A)^{-1} A^*$$

where A^* is the adjoint operator of A. By definition of the adjoint operator, we have for $x \in \mathbb{R}^m$ (with usual inner product) and $y \in \mathbb{R}^N$ (with inner product given by G),

$$\langle Ax, y \rangle_G = \langle x, A^* y \rangle_{\mathbb{R}^m}$$

which gives

$$A^* = A^t G$$

So,

$$\left(J_G F(x)^t\right)^* = J_G F(x) G(x) = JF(x)$$

and the pseudo-inverse is given as

$$\left(J_G F(x)^t\right)^\dagger = \left(JF(x) G(x)^{-1} JF(x)^t\right)^{-1} JF(x) \tag{5.13}$$

These three projection steps are repeated for every iteration of the curve evolution.

5.3 Log map using Parallel transport

Although Log map may not be required for evolving curves, it is necessary for computing geodesics between two given shapes.

While exponential maps are defined by initial value problems (IVP), the computation of geodesic between two points is a Two-Points boundary value problem (TPBVP) and shooting methods [40, 41] have been developed to solve these problems by transforming TPBVPs into IVPs, by an iterative procedure that refines estimates of initial values so that the resulting IVP solution will reach the boundary value.

This is what we will use here in order to compute the Log map between points in our implicitly defines space equipped with a non-Euclidean metric. The procedure consists of updating an initially chosen tangential direction with a vector obtained by parallel transporting an approximate error vector between the current point (obtained by the Exp map of the tangent direction in the current iterate) and the target point. A typical step is illustrated in Figure 5.1 where the geodesic in V joining points q and $\tilde{q}$ is sought.

We describe the step on greater details. First the notations used in the figure. To avoid ambiguities, let Exp_{S^N} denote the exponential map in S^N and Exp_V the exponential in submanifold V. Similar notations Log_{S^N} and Log_V are used for the Riemannian Log maps. A geodesic joining points p and p' in S^N is denoted by $\gamma_{S^N}(p, p')$.

Assume then that $\mathbf{v}^n$ is the current estimate of the needed initial velocity to reach $\tilde{q}$ in V and $q^n = Exp_{V,q}\mathbf{v}^n$ its exponential. If $q^n \neq \tilde{q}$ we want to build an update of the initial velocity $\mathbf{v}^n$ that takes into account the estimation error between q^n and $\tilde{q}$. We approximate this error in several steps. First we consider the geodesic $\gamma_{S^n}(q^i, \tilde{q})$ in S^M linking q^i and $\tilde{q}$ and the log map $\mathbf{u}^n = Log_{S^n,q^n}\tilde{q}$. If the extrinsic curvature of V in S^N is not too "large", the orthogonal projection of $\mathbf{u}^n$ on $T_{q^n}V$ provides an approximation of the V-Log map $Log_{V,q^n}\tilde{q}$. This however requires the computation of $\gamma_{S^n}(q^n, \tilde{q})$, which is a second-order TPBVP, but can be heavy to solve. Instead, assuming that the intrinsic curvature of S^N in a neighborhood of q^n containing $\tilde{q}$ is not to large, making the metric slow varying, we can approximate $\gamma_{S^n}(q^n, \tilde{q})$ by the straight-line segment (the dotted line of the figure) joining q^n and $\tilde{q}$. We then project the vector $\overrightarrow{q^n\tilde{q}}$ to $T_{q^n}V$, this projection being denoted by $\mathbf{w}^n$. Now that we have an estimate of the error at q^n on the form of the vector $\mathbf{w}^n$, we need to carry this information back to the initial location q. This is done by parallel transporting it back along the geodesic $\gamma_V(q, q^n)$ in V joining q and q^n. The parallel transport $P(\mathbf{w}^n)$ is computed via the covariant derivative $\frac{D^V}{dt}$ of $\gamma_V(q, q^n)$ in V. Once done, we update the initial velocity $\mathbf{v}^{n+1} = \mathbf{v}^n + P(\mathbf{w}^n)$.

Such a procedure is generally not guaranteed to converge, it will depend on the intrinsic curvatures of S^N and V in a neighborhood of q and $\tilde{q}$ as well as the extrinsic curvature of V in S^N (generalizing the second fundamental form). When manifolds are sufficiently "flat" it generally work well however.

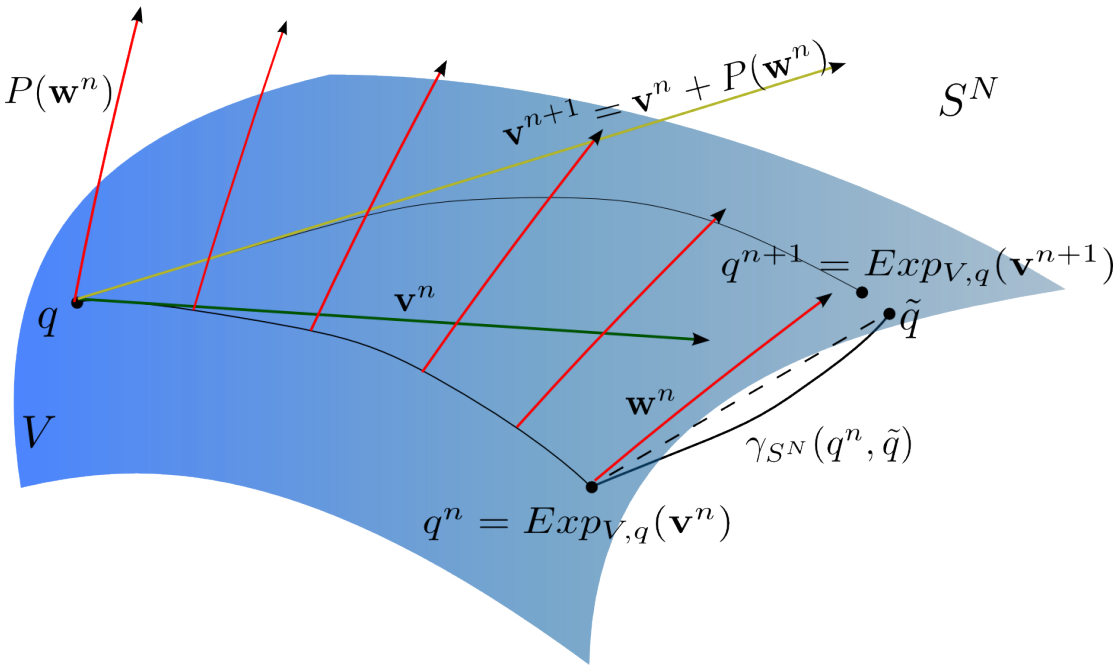


Figure 5.1: A step in Log map / geodesic computation by a shooting method. q is the start point and $\tilde{q}$ the end point ones tries to reach. See the text for further explanations.

The algorithm is given in Table 6.3.1. A fundamental step is computing the parallel transport of a vector $\mathbf{w}(0) \in T_{x(0)}V$ along a geodesic curve $x(t) \in V$. We now derive the needed equations. The

parallel transport is characterized by

$$\mathbf{w}(t) \in T_{x(t)}V \quad \Rightarrow \quad DF(x(t), \mathbf{w}(t)) = 0 \tag{5.14}$$

$$\frac{D^V}{dt}\mathbf{w}(t) = 0 \quad \Rightarrow \quad \frac{D^S}{dt}\mathbf{w}(t) \in N_{x(t)}V \tag{5.15}$$

where $\frac{D^V}{dt}$, $\frac{D^S}{dt}$ are the covariant derivatives with respect to the space V and the embedding space S^N respectively. Differentiating Equation (5.14) and re-arranging terms we get

$$\dot{\mathbf{w}}(t) = -(JF(x))^\dagger \left(\left(D^2F(x)\dot{x}\right)\mathbf{w}\right) \tag{5.16}$$

where $JF(x)^\dagger$ is the G-pseudo-inverse of $JF(x)$ and $D^2F(x)$ is the second differential of F at x. $Jf(x)^\dagger$ is computed using the method described in previous section, which gives

$$Jf(x)^\dagger = G^{-1}JF(x)^t(JF(x)G^{-1}JF(x)^t)^{-1}$$

Equation (5.15) gives

$$\begin{aligned}\frac{D^S}{dt}\mathbf{w}(t) &= \sum_k \left(\frac{dw_k}{dt} + \sum_{ij}\Gamma^k_{ij}w_i x_j\right) \in N_x V \\ &= JF(x)^*\lambda(t)\end{aligned}$$

where Γ are the Christoffel symbols of the embedding space S^N and the adjoint $JF(x)^*$ is given as,

$$JF(x)^* = G^{-1}JF(x)^t$$

Let $f(\mathbf{w}(t), \Gamma, \dot{x}(t)) = \left[\sum_{ij}\Gamma^k_{ij}w_i x_j\right]_{k=1,\ldots,N}$. Rewriting, we get

$$\dot{\mathbf{w}}(t) + f(\mathbf{w}(t), \Gamma, \dot{x(t)}) = JF(x)^*\lambda(t) \tag{5.17}$$

Put Equation (5.16) in Equation (5.17) to compute λ:

$$\lambda(t) = (JF(x)^*)^\dagger \left(f(\mathbf{w}(t), \Gamma, \dot{x}) - JF(x)^\dagger \left(\left(D^2F(x)\dot{x}\right)\mathbf{w}\right)\right)$$

One can then compute $\dot{\mathbf{w}}(t)$ as

$$\dot{\mathbf{w}}(t) = -f(\mathbf{w}(t), \Gamma, \dot{x}) + (JF(x)^*)(JF(x)^*)^\dagger(f(\mathbf{w}(t), \Gamma, \dot{x}) - Jf(x)^\dagger \left(\left(D^2F(x)\dot{x}\right)\mathbf{w}\right)) \tag{5.18}$$

We use a finite difference scheme to implement the above equation.

Compute $Log_{x_1}(x_2)$

1. Set $v = \Pi^G_{T_{x_1}V}(x_2 - x_1)$
2. Compute $\bar{x} = Exp_{x_1}(v)$
3. If $|x_2 - \bar{x}| < \epsilon$, return v, else continue.
4. Compute $\bar{v} = \Pi^G_{T_{\bar{x}}V}(x_2 - \bar{x})$
5. Compute v' the Parallel Transport of $\bar{v}$ to $T_{x_1}V$, using Equation (5.18).
6. Update $v = v + v'$ and go to Step 2.

5.3.1 Stereographic projection

We now implement the Exponential map on a toy example and verify the results before moving on to the more complicated curve evolution equations with the L_2 metric. For this example, we have $S^N = \mathbb{R}^3$, $V = F^{-1}(0), F :$
$setR^3 \to \mathbb{R}$. The metric G is the one induced by stereographic projection $P : \mathbb{S}^3 \backslash (0,0,0,1) \to \mathbb{R}^3$ which is given by

$$P(x_1, x_2, x_3, x_4) = \left(\frac{x_1}{1 - x_4}, \frac{x_2}{1 - x_4}, \frac{x_3}{1 - x_4} \right).$$

Set $\ell := 1 + y_1^2 + y_2^2 + y_3^3$. Then the inverse of P is

$$P^{-1}(y_1, y_2, y_3) = \left(\frac{2y_1}{\ell}, \frac{2y_2}{\ell}, \frac{2y_3}{\ell}, \frac{\ell - 2}{\ell} \right)$$

Then, the corresponding inner product induced on
$setR^3$ at a point $\mathbf{y} = (y_1, y_2, y_3)$ is [47]

$$G_{\mathbf{y}}(v_1, v_2) = \frac{4}{\ell^2} \langle v_1, v_2 \rangle_{\mathbb{R}^3} \tag{5.19}$$

(note that this is a classical example of a conformal equivalence). Let $F_1(\mathbf{y}) = y_1^2 + y_2^2 + y_3^2 - 1$ and $V_1 = F_1^{-1}(0)$. This is the equatorial great cirle of $\mathbb{S}^3$. The gradient and Hessian of F and DG_x are straightforward to compute. Figure 5.2 shows the result of our exponential map algorithm. With this particular choice of F, the situation might be a bit too "toyish". Figure 5.3 shows the exponential map on $V_2 = F_2^{-1}(0)$ for $F_2(\mathbf{y}) = 20y_1 + 2y_2 + 20y_3 - 5y_1^2 - 5y_2^2 - 5x_3^2 - 5$, which is the intersection of a tilted plane passing throught the origin and $\mathbb{S}^3$.
We can verify whether the obtained curves are geodesics or not by checking the tangential component of the acceleration vector. The acceleration vector (in the embedded space) is given by the covariant derivative (for the embedding space) of the velocity vector field (along itself). This is required since the metric on the embedding space is not the usual euclidean one, which gives rise to non-zero Christoffel symbols (Christoffel symbols and covariant derivative definitions are recalled in Appendix A). For a

curve γ of V_i, $i = 1..2$, to be geodesic, we must have

$$\ddot{\gamma}(t) = \nabla_{\dot{\gamma}(t)}\dot{\gamma}(t) \in \mathcal{N}_{\gamma(t)}V_i$$

where $\mathcal{N}_{\gamma(t)}V_i$ is the orthogonal complement of $T_{\gamma(t)}V_i$ in $T_{\gamma(t)}S^N$. The tangential component of the acceleration in V_i is computed by projecting the covariant derivative $\frac{D\dot{\gamma}}{dt}$ of the velocity vector $\dot{\gamma}$ to the tangent space. Figure 5.4 and Figure 5.5 show the norm of different accelerations and projected accelerations for the two geodesics computed on V_1 and V_2 respectively. We also found that the error between the analytically computed geodesic and the geodesic computed with our algorithm reduces as we reduce the time-step, as shown in Figure 5.6. We have also verified our algorithm for Log map on V_1, V_2. In Figure 5.7, we demonstrate parallel transport of a vector along a geodesic in V_2 and show its normal component and covariant derivative in Figure 5.8.

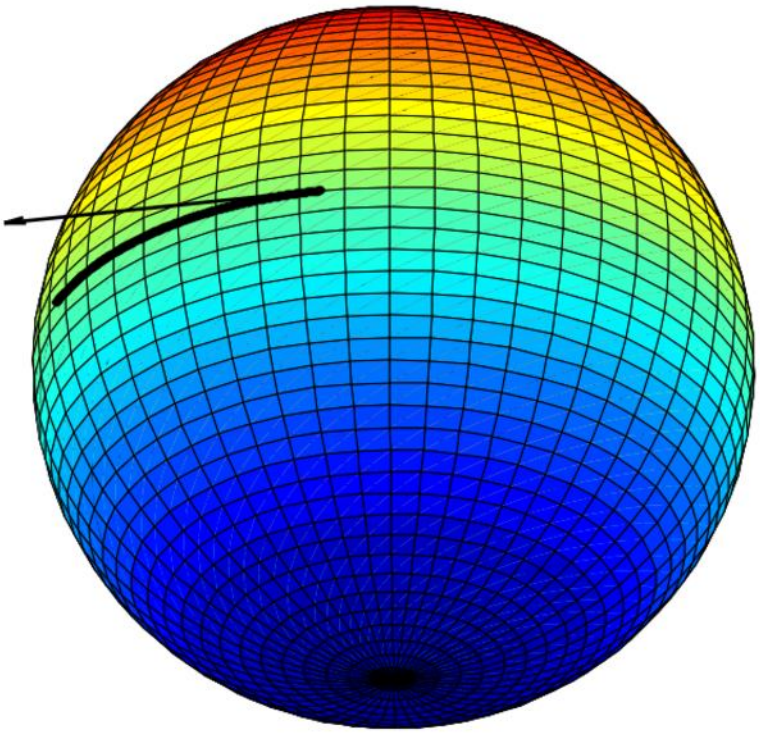


Figure 5.2: Exponential map at $x = (0, 1, 0)$ for $v = (1, 0, 0)$ on V_1

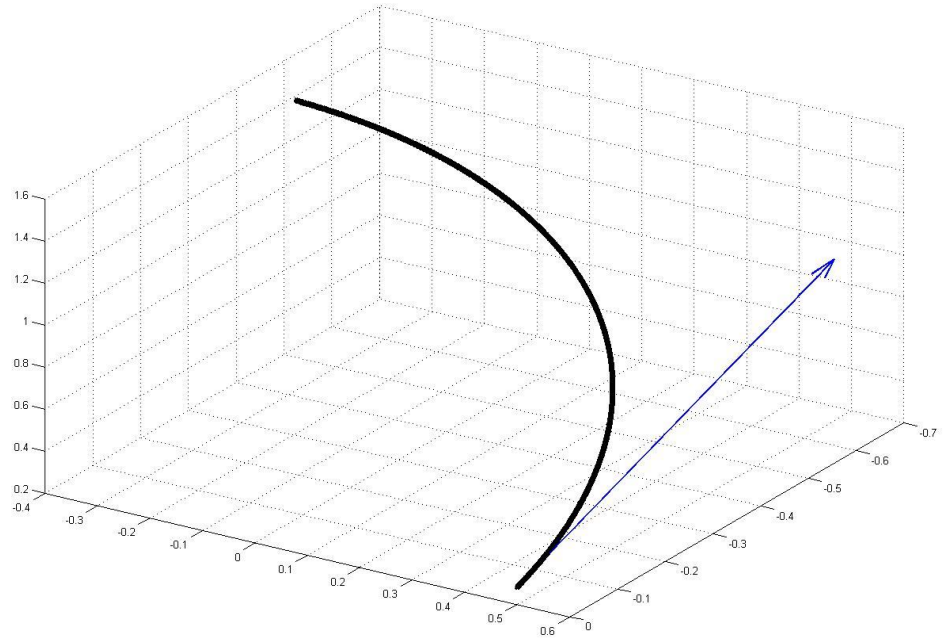


Figure 5.3: Exponential map at $x = (0, 0.5, 0.282444)$ for $v = (-0.7039, 0.03, 0.8249)$ on V_2

5.4 Subspace of Equidistant neighboring node point spline curves

We recall first a few elementary facts on cardinal B-splines, and refer to [82, **?**, 83, 84] for details.

Set $\beta^0(x) = \chi_{[-\frac{1}{2},\frac{1}{2}(}$ the 0-th order basis spline, the n-th order B-splines is defined inductively as

$$\beta^n(x) = \underbrace{\beta^0(x) * \beta^0(x) * \cdots * \beta^0(x)}_{n+1 \text{ times}}$$

where $*$ denote the convolution product.

Fixing such an order and an integer $N > 0$, we consider the space of curves

$$c(s) = \sum_{i \in \mathbb{Z}} \beta^n(s - i)c_i, \quad c_i = (c_i^x, c_i^y) \in \mathbb{R}^2$$

with the periodicity condition

$$c_{i+N} = c_i.$$

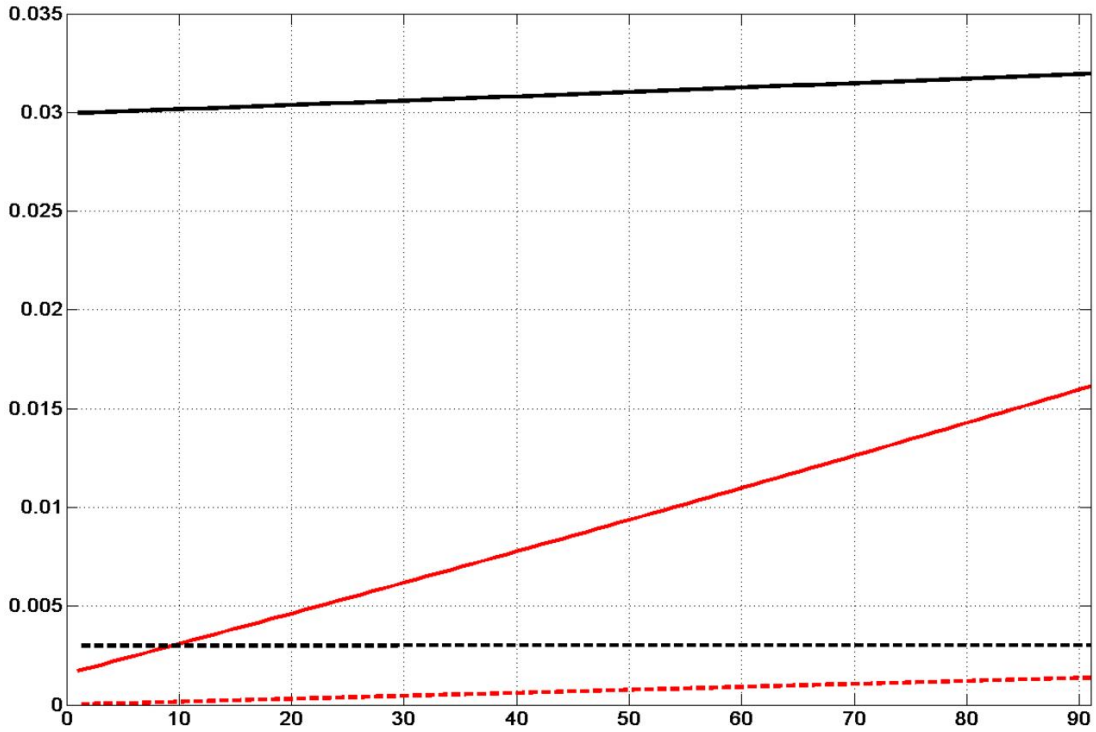


Figure 5.4: The black and red curves are the norm of tangential and normal acceleration vector of the geodesic on V_1, respectively. The solid curves are computed with a time step of 0.01, while the dashed curves are with a time step of 0.001. Due to numerical errors, the tangential component is larger than the normal component of acceleration. Still the magnitudes are the order of 10^{-2}. One can also observe that the tangential and normal component reduce as we reduce the time step, thereby converging to zero, the desired theoretical value.

The periodicity in the sequence of control points insures that the curve c is closed and everywhere of class $\mathcal{C}^{n-1}$. The space of such curves is obviously of dimension $2N$, with the following basis

$$e_{1i} = \begin{pmatrix} \beta^n(s-i) \\ 0 \end{pmatrix}, \; e_{2i} = \begin{pmatrix} 0 \\ \beta^n(s-i) \end{pmatrix}, \quad i = 1, \ldots, N.$$

This is not an orthogonal basis, as, though

$$\langle e_{1i}, e_{2j} \rangle = 0,$$

$$\langle e_{ki}, e_{kj} \rangle = \int_0^N \beta^n(s-i)\beta^n(s-j)\, dx = \delta_{n,|i-j|}$$

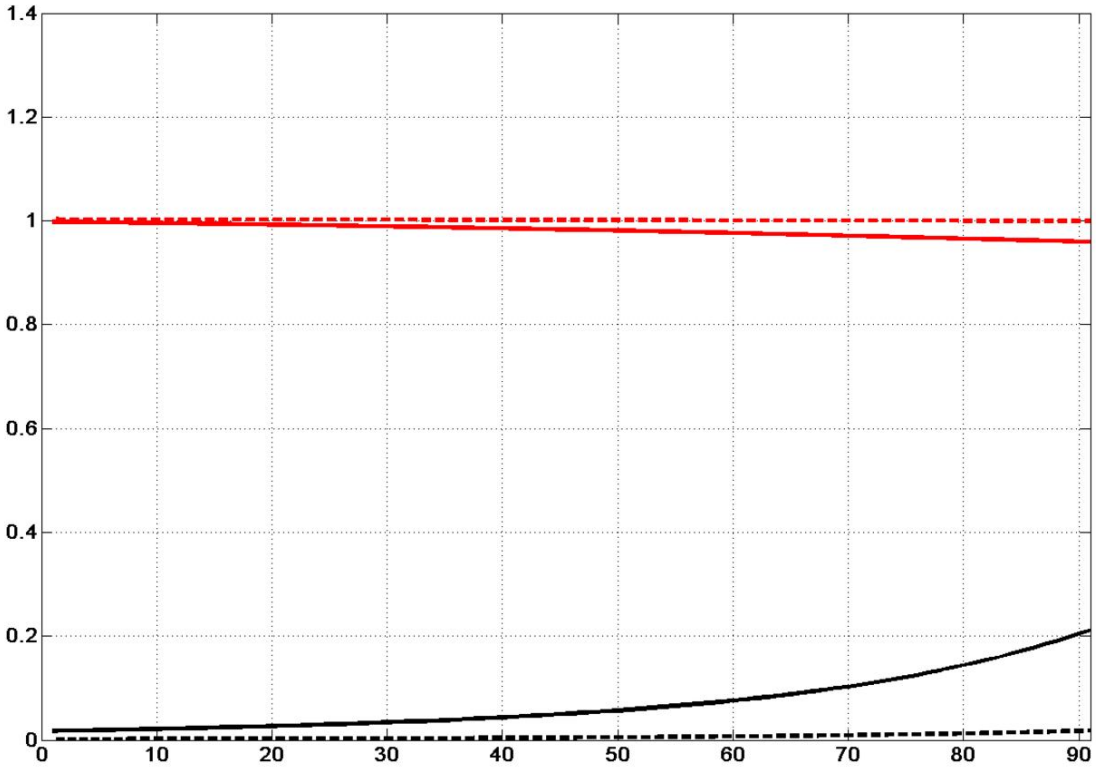


Figure 5.5: The black and red curves are the norm of tangential and normal acceleration vector of the geodesic on V_2, respectively. The solid curves are computed with a time step of 0.01, while the dashed curves are with a time step of 0.001. One can observe that the tangential component reduces as we reduce the time step of the algorithm.

depends on n and the distance between i and j. $(e_{ki})_{i=1,\dots,N}^{k=1,2}$ is not orthogonal with respect to the L^2 inner-product.

Given such a curve c, at each knot $j \in \{1,\dots,N\}$, corresponds a "node point" $P_j = (P_j^x, P_j^y) = c(j)$, and this correspondance is *unique and linear*, given by a $N \times N$ matrix M_N^n

$$\begin{pmatrix} c_1^x & c_1^y \\ \vdots & \vdots \\ c_N^x & c_N^y \end{pmatrix} = M_N^n \begin{pmatrix} P_1^x & P_1^y \\ \vdots & \vdots \\ P_N^x & P_N^y \end{pmatrix}$$

see [3] for details.

We generally start with initial node points and build spline curves passing through them, curves that then will evolve. In most of the curve evolution algorithms we will discuss, we need to be able to compute first and second order derivatives of our spline curves, thus β^n should be of order at least 3. In the sequel

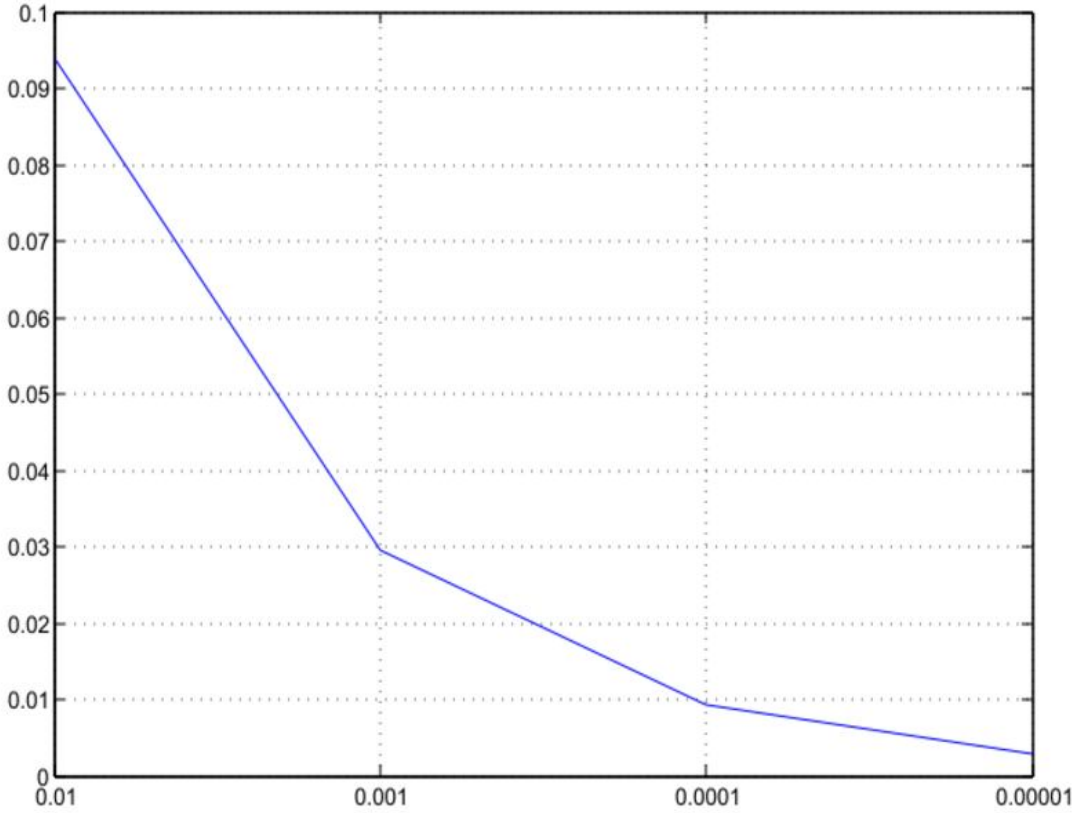


Figure 5.6: Plot of sup norm of the difference between the numerically computed geodesic shown in Figure 5.2 and analytically computed geodesic on V_1 against the time-step used in our algorithm

we assume the order n fixed as well as the number N of control points / node points. This defines a uniqe space

We now define the constraints, as follows. Set

$$d_{i,j} = \|P_j - P_i\|_2^2$$

the squared Euclidean distance between two points in
$\mathbb{R}^2$. Then

which can be written as a simple quadratic constraint $F : S^{2N} \rightarrow$
$\mathbb{R}^{N-1}$ given as

$$\begin{aligned} F_i(P_1, ..., P_N) &= d_{i+2,i+1} - d_{i+1,i}, \;\; i = 1, .., N-2 \\ F_{N-1}(P_1, ..., P_N) &= d_{1,N} - d_{N,N-1} \end{aligned} \tag{5.20}$$

where is the squared euclidean distance between points P_i and P_j, The configuration space is the sub-

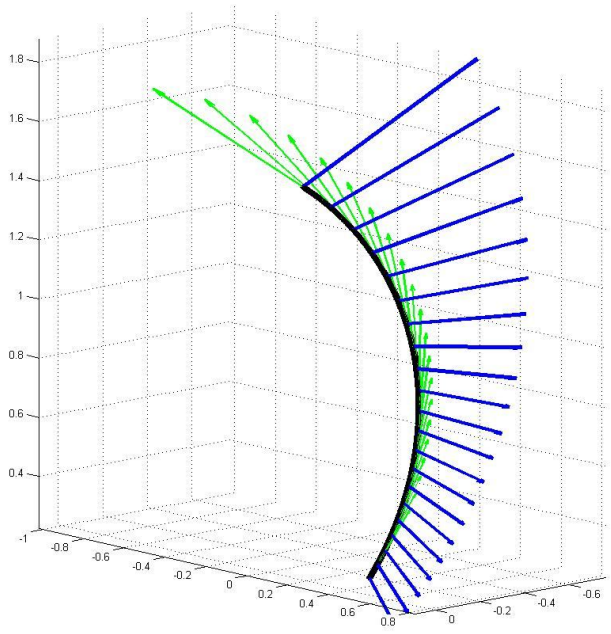


Figure 5.7: Parallel transport of a vector $w \in T_{x(0)}V$ along previously computed geodesic on V_2. The transported vectors are shown in blue and tangent to the geodesic in green

space of S^{2N} given by $\tilde{A}_N = F^{-1}(0)$. The tangent space of $\tilde{A}_N$ at a configuration x is given by

$$T_x\tilde{A}_N = \ker(JF(x)) \tag{5.21}$$

the kernel (or null space) of the Jacobian of F at point $x \in \tilde{A}_N \subset S^{2N}$. We may call $\tilde{A}_N$ a N**-links bicycle chain manifold**. We have explored it in a previous paper [76]. As mentioned in section 5.2, the hierarchy of the subspaces in this case is more complicated. Not all curves in the space of closed spline curves with N control points (S^{2N}) are immersions. Moreover the group of circular permutations on node points has no 'geometric' effect on the spline curve. This will result in a non-zero distance between a curve and a curve obtained by circular permutation of the set of node points of the original curve. We overcome this problem simply by choosing a corresponding starting point on the curves. We let $Imm^N = S^N \bigcap B_i$. Similary not all curves in $\tilde{A}_N$ are immersions. Also the Jacobian of F may not be full rank everywhere on S^{2N}. We let $A'_N \subset \tilde{A}_N$ be the subset where the Jacobian is full rank and let $A_N = A'_N \bigcap Imm^N$ be the space of equidistant neighboring node point closed spline curves that are

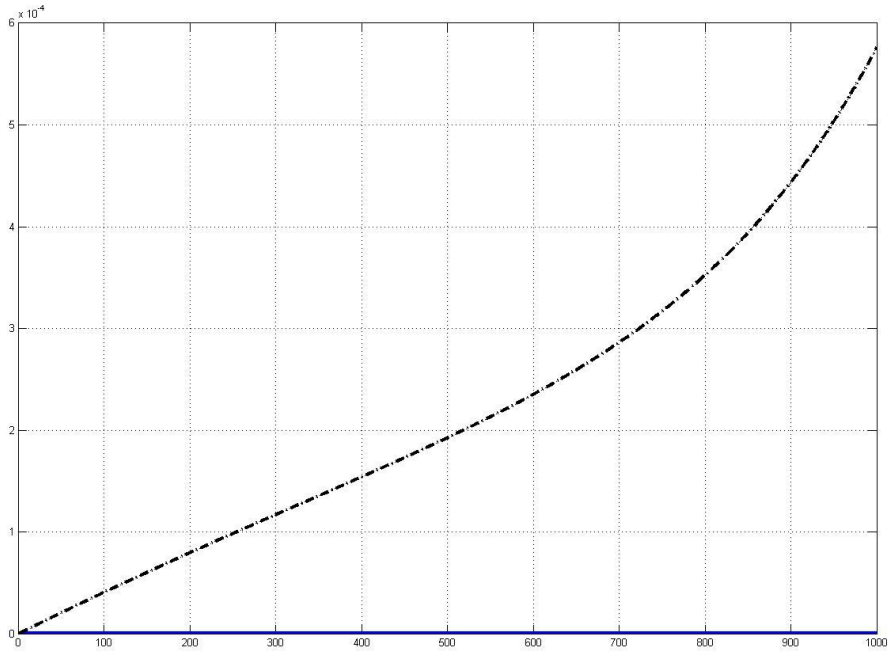


Figure 5.8: The norm of the normal component of the parallel transported vector shown in Figure 5.7 is the black curve ($\simeq 10^{-4}$) and covariant derivative in V_2 of the parallel transported vector is shown in blue ($\simeq 0$)

immersions and where the Jacobian is full rank. For now, given a curve in A_N, we hope that we stay in A_N along an evolution, without putting up additional constraints to ensure that a curve remains an immersions and that the rank of the Jacobian does not reduce (we use $T_x A_N = ker(JF(x))$), accepting the fact that time and again we may end up in the space $\tilde{A}_N \setminus A'_N$.

We now need to construct an inner product on A_N. Instead of restricting the scalar product of $\mathbb{R}^{2N}$ to $T_x A_N$, we induce the L_2 metric of B_i and S^N on $T_x A_N$.* We now derive the expression for the L_2 inner product on the spline subspace. Given two deformations $p(s) = \sum_i \beta^n(s-i)p_i$ and

*Not all N node points spline curves are immersions. But we assume that the evolution velocity keeps the curve in B_i

$q(s) = \sum_i \beta^n(s-i)q_i$ of the spline curve $c(s)$, the inner product of the deformations is given as

$$\begin{aligned}\langle p(s), q(s)\rangle_c &= \int_0^1 \left(\sum_i \beta^n(s-i)p_i\right)\cdot n(s) \\ &\quad \left(\sum_j \beta^n(s-j)q_j\right)\cdot n(s)|c'(s)|\,ds \\ &= P^t G(c) Q\end{aligned}$$

where $a \cdot b$ is the usual dot product between two vectors a and b, $n(s)$ is the inner unit normal to the curve $c(s)$ and $P = (p_x^0, p_y^0, \ldots, p_x^{N-1}, p_y^{N-1})$ and $Q = (q_x^0, q_y^0, \ldots, q_x^{N-1}, q_y^{N-1})$ are the control point vectors. The Gram matrix $G(c)$ is a $2N \times 2N$ matrix which can be written as an $N \times N$ matrix of smaller 2×2 matrices given by

$$G_{ij}(c) = \int_0^1 \beta^n(s-i)\beta^n(s-j)n(s)n(s)^t|c'(s)|\,ds$$

One can clearly see the smoothing effect of the splines in the Gram matrix. This limits the high frequency variation of a deformation. The derivative of the Gram matrix with respect to the node points C_k of the curve $c(s)$ at which it is computed is a $2N \times 2N \times 2N$ matrix whose members are $2N \times 2N$ matrices of the form

$$\frac{dG_{ij}}{dC_k}(c) = \int_0^1 \beta^n(s-i)\beta^n(s-j)\frac{d}{dC_k}(n(s)^t n(s)|c'(s)|)\,ds \tag{5.22}$$

The inner unit normal of a curve $c(s) = (x(s), y(s))$ is given as $n(s) = \frac{(-y'(s), x'(s))}{\sqrt{x'(s)^2 + y'(s)^2}}$. Given the direct and indirect spline transforms [83, 84], one can compute the derivatives in Equation (5.22) with little work. The gradient and Hessian of the equidistant constraint are easy to compute and the pseudoinverse of the $J_G F^t$ matrix can be computed as shown in Equation (5.13).

To restrict a given curve evolution to the subspace A_N, we simply iterate the three projection steps described in section 5.2. The projection to S^{2N} is given by (using Equation (5.4))

$$v_S(t) = \sum_j \left[G_1^{-1}\left(\int_0^L e_i(s)v(s,t)n(s,t)\,ds\,|_{i=1\ldots N}\right)\right]_j e_j \tag{5.23}$$

where e_i are the spline basis vectors of S^{2N}. One can observe that computation of the motion of a control point due to the velocity $v(s,t)n(s,t)$ is non local, i.e., depending on the basis vectors (splines in this case), the control point motion computation is a weighted average like process over a certain neighborhood. This comes from the fact that finite dimensional curve representation has to be ¡non-local in nature. The spline basis vectors filter out high frequency variations of the velocity field on the curve, which is in some sense gives similar effect to using the Sobolev inner product on the space of curves (as in [78]). The projection to $T_{c(t)}A_N$ is given by Equation (5.6) and we can further project the obtained tangent velocity vector to the N-links bicycle chain manifold using the Exponential map given

by Equations (5.10), (5.11), (5.12) and (5.13).

5.5 Experiments

We restrict curvature flow and geodesic active contour evolution equation to the N-links bicycle chain manifold defined in the previous section. Curvature flow is given as,

$$\frac{\partial C}{\partial t}(p,t) = \kappa(p,t)n(p,t) \tag{5.24}$$

where κ is the curvature.

Curve evolution equation for the geodesic active contour segmentation model is given as,

$$\frac{\partial C}{\partial t}(p,t) = (\kappa g - \langle \nabla g, n \rangle + \alpha g)n \tag{5.25}$$

where g :
$\mathbb{R} \rightarrow$
$\mathbb{R}$ is the edge detection function

$$g(|\nabla I(C(p))|) = \frac{1}{(1+|\nabla I(C(p))|)^2}$$

and the αg term helps in detecting non-convex objects [1]. Figure 5.9 shows the curvature flow of a non-convex curve restricted to the N-links bicycle chain manifold. Observe that there is a tangential motion to the curve in order to satisfy the implicit constraints. We give a comparison between curvature flow restricted to the N-links bicycle chain manifold and to the finite dimensional linear spline subspace in Figure 5.10.

In Figure 5.11 we demonstrate segmentation on a heart image with geodesic active contours restricted to the N-links bicycle chain manifold. The Log map result is shown in Figure

5.6 Conclusion and Future work

In this paper we give a new algorithm to restrict curve evolution to implicitly defined subspace induced with a non-euclidean metric. We give a new algorithm to compute the Exponential map on implicitly defined manifolds equipped with a non-Euclidean metric. The N-links bicycle chain manifold is a useful space to work in, specially in medical imaging, where landmarks and pseudo-landmarks are often used. Moreover this algorithm can be used as an optimization method to optimize over implicitly defined manifolds with a given inner product.

One of the drawbacks of this method is that as the number of points on the curve increases, computing the derivative of the Gram matrix $\frac{dG}{dX}$ becomes very expensive. The next obvious step is to compute the Log map with non-Euclidean inner product on the proposed shape space, so that geodesic distance can

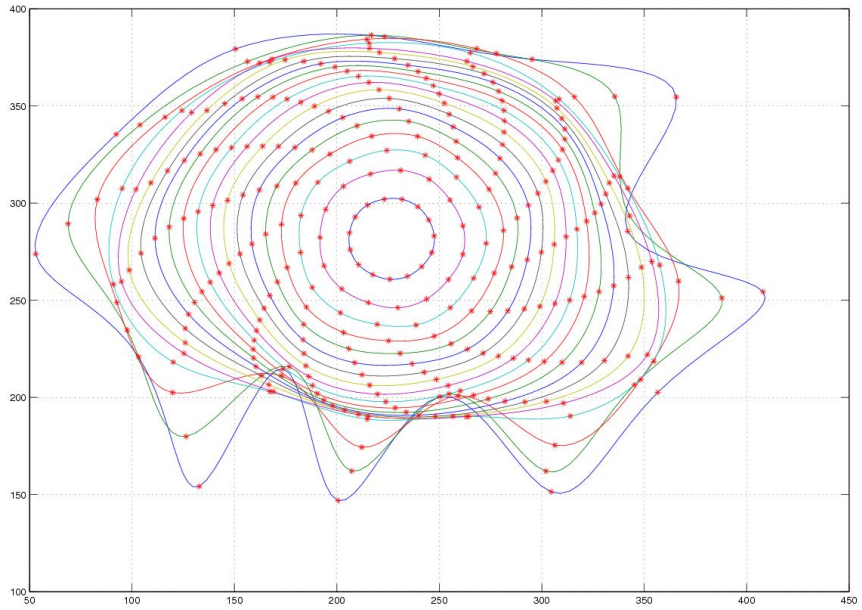


Figure 5.9: Curvature flow of a non-convex curve. The points marked with red '*' are the node points. Observe that there is a tangential motion to the curves in order to satisfy the equidistant constraint

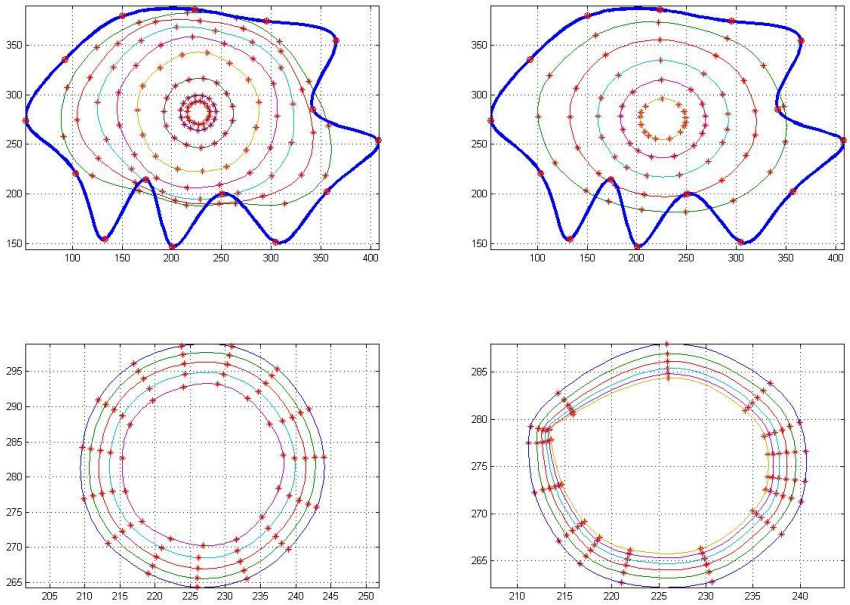


Figure 5.10: (top-left) Curvature flow restricted to N-links bicycle chain manifold. (top-right) Curvature flow restricted to linear spline subspace. (bottom-left) some of the final iterations of curvature flow in N-links bicycle chain manifold. (bottom-right) some of the final iterations of curvature flow in spline subspace. Notice the numerical errors arising because of node points coming too close to each other

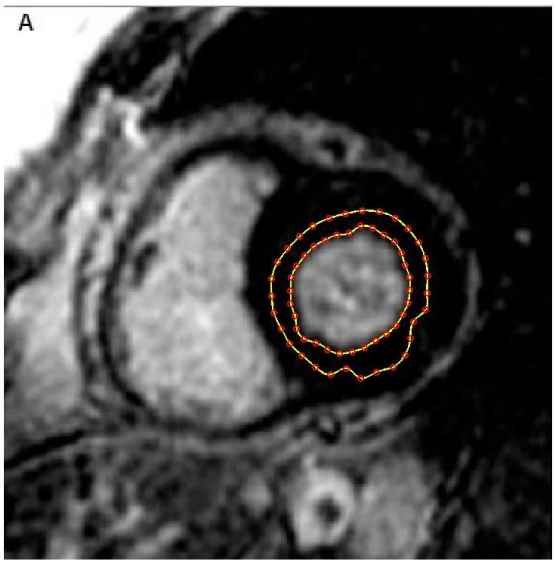


Figure 5.11: Geodesic active contours on a heart image. The points marked with red 'o' are the node points.

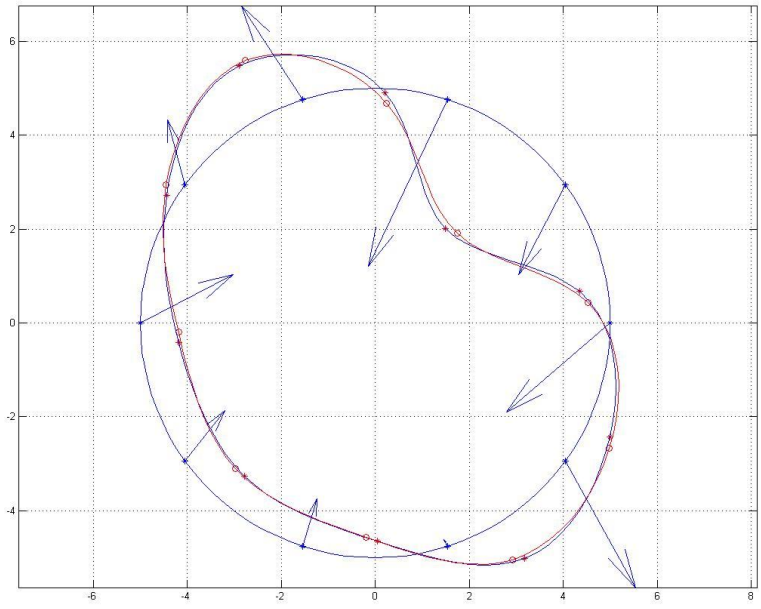


Figure 5.12: Log of the deformed blue curve (with node points marked by '*') with a circle (in blue) as the base point is shown by deformation vectors on the node points of the circle. The red curve (with node points marked with 'o') is the Exponential map of the computed deformation field

be computed for any two given shapes.

Chapter 6

Metameric class of Histogram of Gradient Orientation features

We plan to submit this chapter as *Metameric class of Histogram of Gradient Orientation features* to *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

6.1 Introduction

Object recognition is a vital component of a general computer vision system. For this task numerous image features have been developed that can identify the object under different transformations. The appearance of objects changes due to transfomations like change in viewing angle, illumination changes, contrast change, noise etc. Features must be invariant to most of these transformations so that the object can be identified in different images. On the other hand, different objects should have different features so that the system is able to differentiate between them. There is therefore, a trade-off in what the feature should preserve and what it should overlook.

Some of the more successful features are based on the concept of Histogram of Gradient orientation (henceforth written as HOGO). Many versions of this feature have come up with different modifications. Edge orientation histograms [79], shape contexts [2], SIFT [48], PCA-SIFT [86], GLOH [53], Histogram of Oriented Gradients [17] are few of them. A comparative study of the performance of some of the above mentioned features and others is given in [53]. In most of the above cases, the HOGO is computed in a neighborhood(image patch) of some points of the image.

We choose to work with the basic HOGO feature (description of the feature in next section) and try to explore the space of image patches with the same feature. Ideally the space should neither be too big (many objects might have the same feature) nor too small (same object under some transformation may have different feature). There are many image patches having a particular HOGO feature. some of them might be simply scaled versions of the original patch, while some might appear meaningless (noise-like). We propose a method to sample a 'meaningful' patch from the distribution specified by the

HOGO feature. Given two image patches u and w defined on a common domain Ω, our method generates a patch that looks like w (in the L^2 sense) but has the same HOGO feature as u.

In this paper, we give a method to evolve a patch u towards an approximation (or projection) of another given patch w preserving the HOGO feature of u. In [50], Lillholm *et al.* generate an image patch which is the simplest possible (in some metric), with the same linear scale space features as a given patch. We extend the study to HOGO features which are nonlinear. This is also a first step towards understanding why these features work as well as they do.

In Section 6.2, we descibe the particular feature we are dealing with and explain its metameric class. We model our problem as a constrained optimization problem and give the theoretical and numerical methods we use to solve it in Section 6.3. We show some of our results in Section 6.4 and conclude in Section 6.5.

6.2 Equivalence class of HOGO features

We start by defining our HOGO feature. We use a 8 bin angular histogram with bins centered at a distance of 45° starting at 0°. The HOGO of an image patch u defined on $\Omega \subset \mathbb{R}^2$ is defined as

$$\begin{aligned} h_{\theta_i}(u) &= \int_\Omega A_{\theta_i}(u) dx\, dy \\ A_{\theta_i}(u) &= \{(x,y) \in \Omega | \theta_i - 22.5^\circ < \tan^{-1}\left(\frac{u_y(x,y)}{u_x(x,y)}\right) < \theta_i + 22.5^\circ\} \end{aligned}$$

with $\theta_i \in (0^\circ, 45^\circ, \ldots, 315^\circ)$ and the spatial derivatives u_x, u_y of u are computed at a certain scale σ_0.

A histogram can also be completely characterized by the statistical moments. A standard result from statistics [59] states that a distribution function is completely determined by all its statistical moments. We can approximate the histogram if we know a finite number of its statistical moments. The p^{th} statistical moment of some directional data ϕ defined over $\Omega \subset \mathbb{R}^2$ is a complex number

$$r_p = \frac{1}{|\Omega|}\left[\int_\Omega \cos(p\phi) dx\, dy + i \int_\Omega \sin(p\phi) dx\, dy\right]$$

We denote the p^{th} statistical moment of the gradient angle of an image patch u as a pair of real numbers

$$G_p^m(u) = \int_\Omega \cos\left(p \tan^{-1}\left(\frac{u_y}{u_x}\right)\right) dx\, dy = u_p^m \in \mathbb{R} \tag{6.1}$$

$$G_p^n(u) = \int_\Omega \sin\left(p \tan^{-1}\left(\frac{u_x}{u_x}\right)\right) dx\, dy = u_p^n \in \mathbb{R} \tag{6.2}$$

We neglect normalization with $|\Omega|$ since we assume that all the given patches are defined on a fixed Ω .

Then the equivalence class of patches with same HOGO* as u is given as

$$M = \{w \in L^2(\Omega) | G_p^m(w) - u_p^m = 0, G_p^n(w) - u_p^n = 0, 1 \leq p \leq k\} HoG \tag{6.3}$$

It is easy to see that the space M is nonlinear. In order to evolve one patch into another on this nonlinear space, we need tools from differential geometry, for example the exponential map. Since the space is an implicitly defined infinite dimensional space, computing the exponential map is not trivial. We approximate the exponential map using the gradient descent method. A vector field is defined as the gradient of the L^2 distance between the two intially given patches. A different way of preserving image histograms is given in [22], where the authors use warping methods on the domain of the image, which we do not want. We now describe our method in detail.

6.3 Constrained optimization

The problem clearly falls into the category of constrained optimization. Given two patches u and w, minimize

$$J(f) = \int_\Omega (u + f - w)^2 dx\, dy$$

where f is a perturbation function which evolves u. Let $g = u + f$ denote the evolving patch. The constraints that g needs to satisfy are given as

$$G_p^m(f) = \int_\Omega \cos(p\theta) dx\, dy - u_p^m = 0 \quad 1 \leq p \leq k \tag{6.4}$$

$$G_p^n(f) = \int_\Omega \sin(p\theta) dx\, dy - u_p^n = 0 \quad 1 \leq p \leq k \tag{6.5}$$

where $\theta = \tan^{-1}\left(\frac{g_y}{g_x}\right)$ and k is some integer.

We form the Lagrange function L with Lagrange multipliers $\lambda_1, \ldots, \lambda_k, \mu_1, \ldots, \mu_k \in \mathbb{R}$:

$$\begin{aligned} L(f, \lambda_1, \ldots, \lambda_k, \mu_1, \ldots, \mu_k) = & \int_\Omega (u + f - w)^2 \\ & - \sum_{p=1}^k \lambda_p \left(\cos\left(p \tan^{-1}\left(\frac{g_y}{g_x}\right)\right) - m_p \right) \\ & - \sum_{p=1}^k \mu_p \left(\sin\left(p \tan^{-1}\left(\frac{g_y}{g_x}\right)\right) - n_p \right) dx\, dy \end{aligned}$$

The gradient of L is

$$\nabla_f L = \nabla_f J - \sum_{p=1}^k \lambda_p \nabla_f G_p^m - \sum_{p=1}^k \mu_p \nabla_f G_p^n \tag{6.6}$$

*We preserve enough moments for a good approximation of the histogram

The corresponding gradients are:

$$\nabla_f J = 2(u + f - w) \tag{6.7}$$
$$\nabla_f G_p^m = \text{div}_x(\eta \sin(\phi)\, \nabla_x g^\perp) \tag{6.8}$$
$$\nabla_f G_p^n = -\text{div}_x(\eta \cos(\phi)\, \nabla_x g^\perp) \tag{6.9}$$

where

$$\eta = \frac{p}{g_x^2 + g_y^2} \quad \phi = p\, \tan^{-1}\left(\frac{g_y}{g_x}\right)$$

We evolve the function f using the gradient descent scheme:

$$\frac{df}{dt} = -\nabla_f L(f(t), \lambda_1(t), \ldots, \lambda_k(t), \mu_1(t), \ldots, \mu_k(t)) \tag{6.10}$$

To use this equation, we need to compute the Lagrange multipliers. We follow the method of Solem and Overgaard [75]. We know that for $t \geq 0$ and for $1 \leq p \leq k$

$$G_p^m(f(t)) = 0 \quad G_p^n(f(t)) = 0$$

Differentiating the first set of constraints with respect to t, we get

$$DG_p^m(f, \dot{f}) = 0$$

where $\dot{f} = \frac{df}{dt}$.

Using Equation (6.10), we get

$$DG_p^m(f, \dot{f} = -\langle \nabla_f G_p^m(f), \nabla_f L \rangle = 0$$

where $\langle -, - \rangle$ is the L^2 inner product. Using Equation (6.6), we get,

$$-\left\langle \nabla_f G_p^m(f), \nabla_f J - \sum_{i=1}^{k} \lambda_i \nabla_f G_i^m - \sum_{j=1}^{k} \mu_j \nabla_f G_j^n \right\rangle = 0$$

Similarly using $G_p^n(f(t)) = 0$, we get

$$-\left\langle \nabla_f G_p^n(f), \nabla_f J - \sum_{i=1}^{k} \lambda_i \nabla_f G_i^m - \sum_{j=1}^{k} \mu_j \nabla_f G_j^n \right\rangle = 0$$

Therefore λ_i, μ_j are the $2k$ constants satisfying the $2k$ equations:

$$\begin{aligned}
\sum_{i=1}^{k} \lambda_i \left\langle \nabla_f G_p^m, \nabla_f G_i^m \right\rangle + \sum_{j=1}^{k} \mu_j \left\langle \nabla_f G_p^m, \nabla_f G_j^n \right\rangle &= \left\langle \nabla_f G_p^m, \nabla_f J \right\rangle \\
& \quad 1 \leq p \leq k \\
\sum_{i=1}^{k} \lambda_i \left\langle \nabla_f G_p^n, \nabla_f G_i^m \right\rangle + \sum_{j=1}^{k} \mu_j \left\langle \nabla_f G_p^n, \nabla_f G_j^n \right\rangle &= \left\langle \nabla_f G_p^n, \nabla_f J \right\rangle \\
& \quad 1 \leq p \leq k
\end{aligned} \tag{6.11}$$

Let $\lambda = [\lambda_1, \ldots, \lambda_k]$, $\mu = [\mu_1, \ldots, \mu_k]$, $(A^{mm})_{i,j} = \langle \nabla G_i^m, \nabla G_j^m \rangle$, $(A^{mn})_{i,j} = \langle \nabla G_i^m, \nabla G_j^n \rangle$, $(A^{nn})_{i,j} = \langle \nabla G_i^n, \nabla G_j^n \rangle$, $(G^m J)_i = \langle \nabla G_i^m, \nabla J \rangle$ and $(G^n J)_i = \langle \nabla G_1^n, \nabla J \rangle$. Rewriting the equations, we get

$$\begin{bmatrix} \lambda & \mu \end{bmatrix} \begin{bmatrix} A^{mm} & A^{mn} \\ A^{mn} & A^{nn} \end{bmatrix} = \begin{bmatrix} G^m J & G^n J \end{bmatrix}$$

Let

$$A = \begin{bmatrix} A^{mm} & A^{mn} \\ A^{mn} & A^{nn} \end{bmatrix}$$

Then,

$$\begin{bmatrix} \lambda & \mu \end{bmatrix} = \begin{bmatrix} G^m J & G^n J \end{bmatrix} A^{-1}$$

which then gives the evolution equation:

$$\frac{df}{dt} = -\left(\nabla_f J - \sum_{i=1}^{k} \lambda_i \nabla_f G_i^m - \sum_{j=1}^{k} \mu_j \nabla_f G_j^n \right) \tag{6.12}$$

The right hand side expression in the above equation is the projection of the gradient ∇J to the tangent spaces of the constraint space $G_i^m, G_i^n = 0$. Since we implement this using discrete time-steps, the constraint value tends to drift away from zero. Hence we need some approximation of the Exponential map to stay on the subspace. We use the gradient descent method used in Chapter 4 for this purpose.

Define $G^k : L^2(\Omega) \to \mathbb{R}$:

$$G_k = \frac{1}{2} \sum_{i=1}^{k} (G_i^m - u_i^m)^2 + \frac{1}{2} \sum_{i=1}^{k} (G_i^n - u_i^n)^2$$

It is obvious that $u \in (G_k)^{-1}(0)$, i.e. it belongs to the zero-level set. The space $L^2(\Omega)$ can be seen as a collection of level sets $(G_k)^{-1}(a), a \in \mathbb{R}$. Let $w \in (G_k)^{-1}(a_w)$. Then we can use a gradient descent scheme starting at $w \in L^2(\Omega)$ to reach the zero-level set of G_k, which will give the required image having the same moments as u, since the space M can also be written as $(G_k)^{-1}(0)$. The gradient descent is given as

$$\frac{dw}{dt} = -\nabla G^k(w(t))$$

It is straighforward to see that

$$\nabla G_k = \sum_{i=1}^{k} (G_i^m - u_i^m) \nabla G_i^m + \sum_{i=1}^{k} (G_i^n - u_i^n) \nabla G_i^n \tag{6.13}$$

The individual gradients of the constraint functions are given in Equation (6.9). We now explain our numerical algorithm to carry out the minimization.

6.3.1 Numerical algorithm

Due to the infinite dimensional and nonlinear nature of the constrained space we are working on, one can imagine that the evolution of a particular patch can get stuck at one of the several local minimas. In order to get rid of this problem, we let the evolution carry on for certain number of iterations instead of looking for minimas. Once we have all the intermediate patches, we choose by hand the appropriate minima. Our purpose here is not to find the global minima, but just to illustrate the possibility of finding a patch that looks like another given patch, having the HOGO feature of the initial patch.

The gradient descent step used to project a point from the tangent space onto the space needs a numerical stopping criteria. We fix a threshold value G_{th} on the level set function G_k, and we allow the gradient descent to reach a local minima. We agree on the local minima, if the function value at the point is below G_{th}. If the function value at the local minima is beyond G_{th}, then we half the timestep on the tangent space and start the gradient descent from the new point. There is a possibility that we might need extremely tiny timesteps at points where the space is highly curved. But in such cases, there is practically no change observed in the patch. To overcome this problem we set a threshold on the tangential timestep τ_{th}. If the timestep goes below τ_{th}, we increase the G_{th} to 1.5 times its current value. Once this particular iteration is done, we reset all the timesteps and threshold values to their original values. The numerical algorithm is summarized in Table 6.3.1 and is also demonstrated in Figure 6.1.

6.4 Experiments

We use different categories of images, one set containing patches of size 50×50 pixels and the other 10×10 pixels. The 50×50 pixel size patches make it easier to judge visual simalarity between patches. The formulation of the method itself makes the algorithm dependant on the size of patches. Higher the size, more the free variables the algorithm has to manipulate the patches. We show some of our results on 50×50 pixel patches in Figures 6.2 and 6.3. We can observe that two completely different looking patches can be made to have same HOGO feature. Thus one should not blindly use the HOGO feature. From our results, we find a general trend that the output of our algorithm is the target image with a noise-like component added to it. We choose 20 random images from the Caltech database and resize them to 50×50 pixels patches. We then evolve a given patch to each of these patches with number of moments preserved varying from 1 to 32. We choose sum of square distance of pixels in a patch as a distance measure. We then compute the ratio of distance between the evolved and target patch to the distance

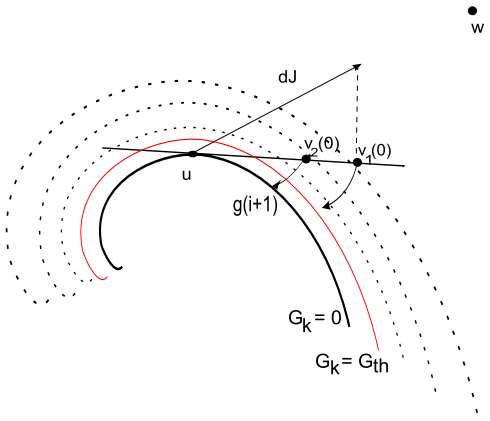


Figure 6.1: An iteration of our algorithm. The gradient descent from point $v_1(0)$ converges to a local minima, say x_1, for which $G(x_1) > G_{th}$. So we half the time step on the tangent space and do a gradient descent from $v_2(0)$ which then converges to the constraint space.

1. Initial patch $g(0) = u$, Target patch w. Set initial parameters: $\tau, \tau_{gd}, G_{th}, \tau_{th}$
2. Loop until certain number of iterations:
   (a) compute Lagrange multipliers λ_i, μ_i for current patch g using Equations (6.11) and compute ∇L, using Equations (6.6).
   (b) Set $w(0) = g(i) - \tau \nabla L$
   (c) Loop untill a local minima patch (say $\tilde{w}$) is obtained:
      i. Compute $w(j+1) = w(j) - \tau_{gd} \nabla G_k$ using Equation (6.13)
   (d) Check $G_k(\tilde{w}) < G_{th}$.
      - If false, set $\tau = \frac{\tau}{2}$
      - Check $\tau < \tau_{th}$.
        – If false go to Step 2b
        – If true, set $G_{th} = 1.5 G_{th}$ and go to Step 2b.
      - If true set $g(i+1) = \tilde{w}$ and increment the iteration counter and reset all parameters to their initial values given in Step 1.
3. Choose $g(i)$ such that $G_k(g(i))$ is as low as possible and $J(g(i) - u)$ is as low as possible.

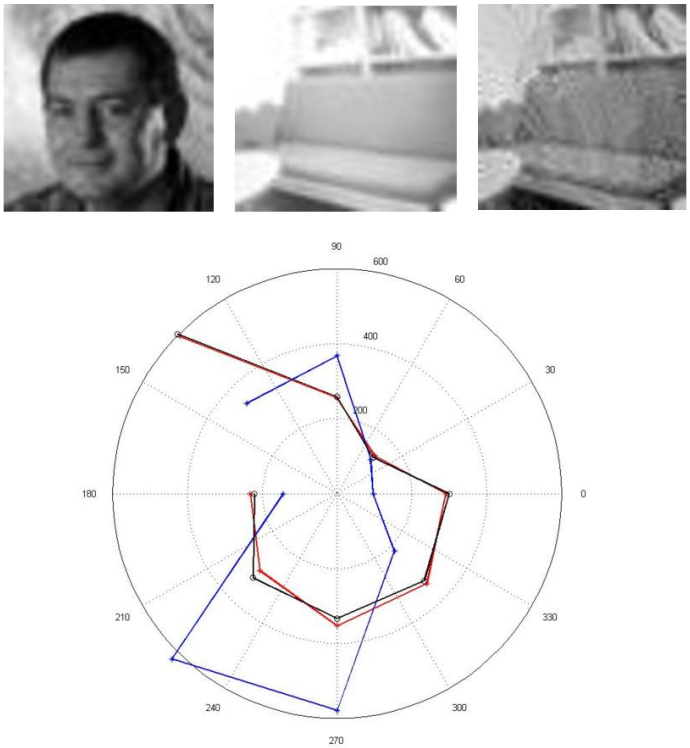


Figure 6.2: (top-left) Initial patch, (top-center) Target patch, (top-right) Output of the algorithm, (bottom) HOGO of the three patches - Initial patch (red), Target patch (blue), Output patch (black with 'o')

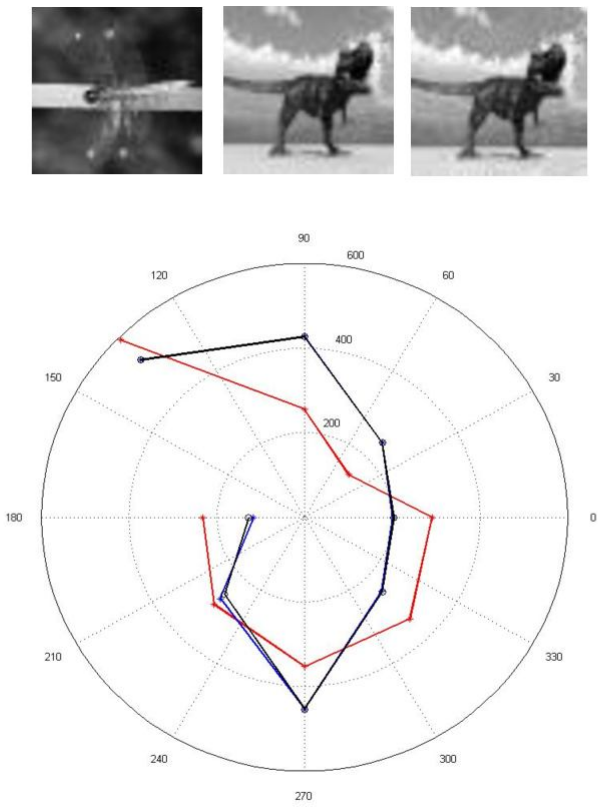


Figure 6.3: (top-left) Initial patch, (top-center) Target patch, (top-right) Output of the algorithm, (bottom) HOGO of the three patches - Initial patch (red), Target patch (blue), Output patch (black with 'o')

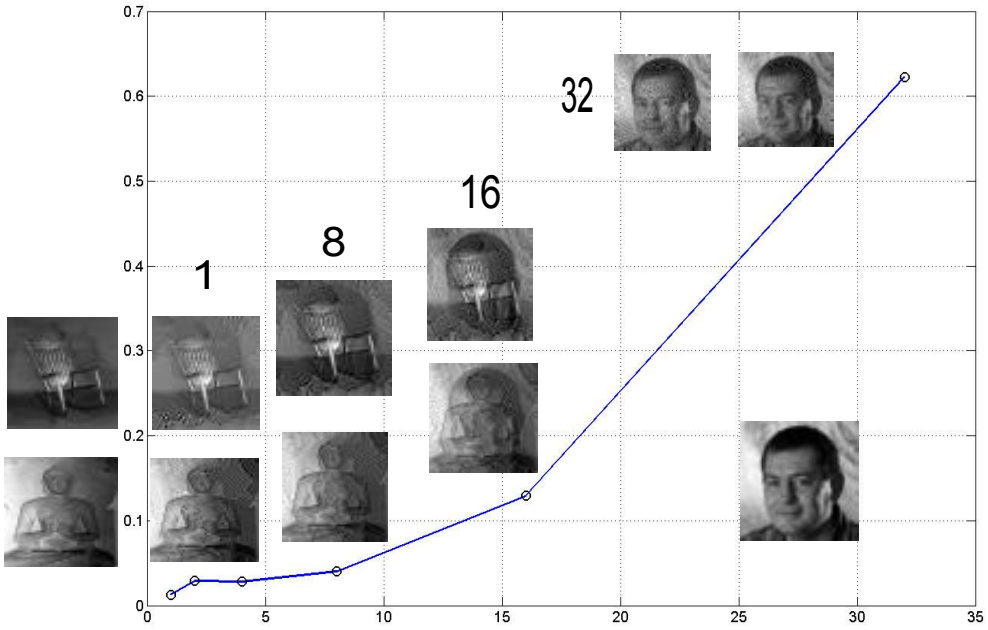


Figure 6.4: The plot show the average ratio of distances between the evolved and target patches to the distance between the initial and the target patch, computed on 20 different target patches with number of moments preserved on the x-axis. To the left of the y-axis are two of the target patches used, while the initial image is shown on the bottom right corner of the plot. Also shown along the plot are examples of evolved patches with number of constraints given on the top.

between the initial and the target patch. Then an average of this ratio is computed for the 20 images separately for each number of constraint considered. We plot this ratio in Figure 6.4. We see that even with 16 moments preserved, the patch can change a lot. We show couple of evolved patches obtained for two of the patches in the database. We show the histograms of the initial, target and evolved patches in Figure 6.5. Similar plots using 15, 10×10 pixel size patches are shown in Figures 6.66.7. In this case, even with 8 moments preserved, on an average one can reduce the sum of squared distance by half. This shows considerable variability in the equivalence class. The histograms are difficult to preserve for smaller patches due to the low number of samples available.

6.5 Conclusion

We devise an algorithm that can evolve one given patch into a completely different looking patch, but still having the same HoG feature as the initial one. This is a way to explore the metameric class of patches

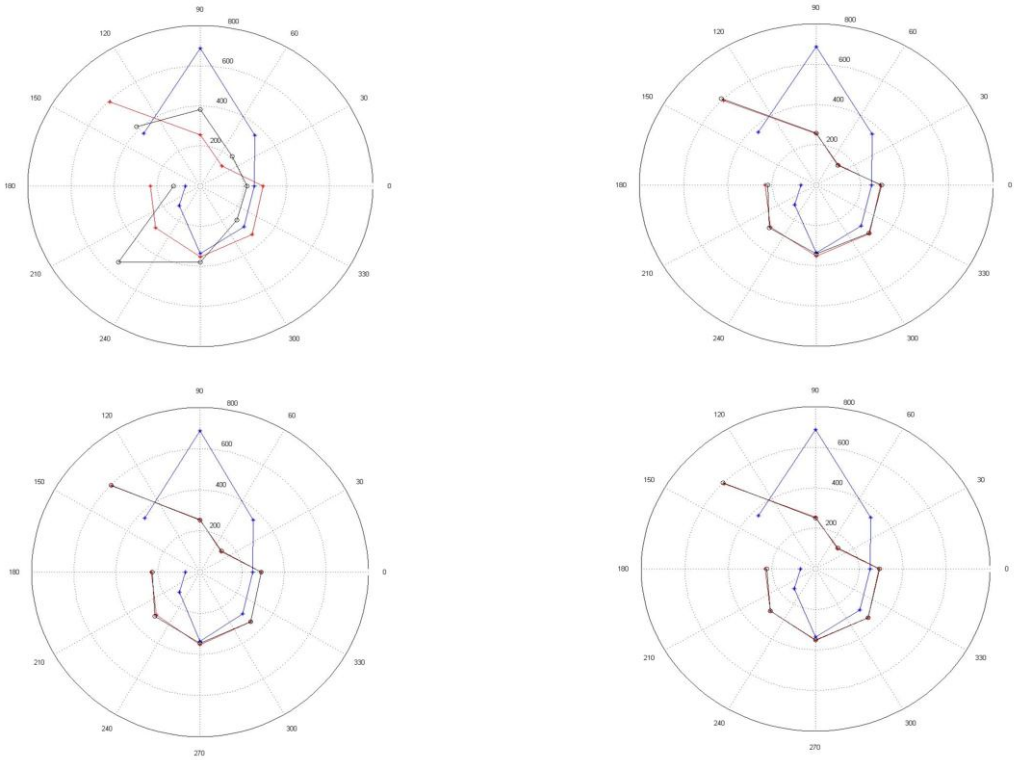


Figure 6.5: The histograms of the initial patch (plot in red with '*'), target patch (plot in blue with '*') and evolved patch (plot in black with 'o') for (top-left) 1,(top-right) 8,(bottom-left) 16 and (bottom-right) 32 number of moments preserved for the rocking chair patch shown in the result before.

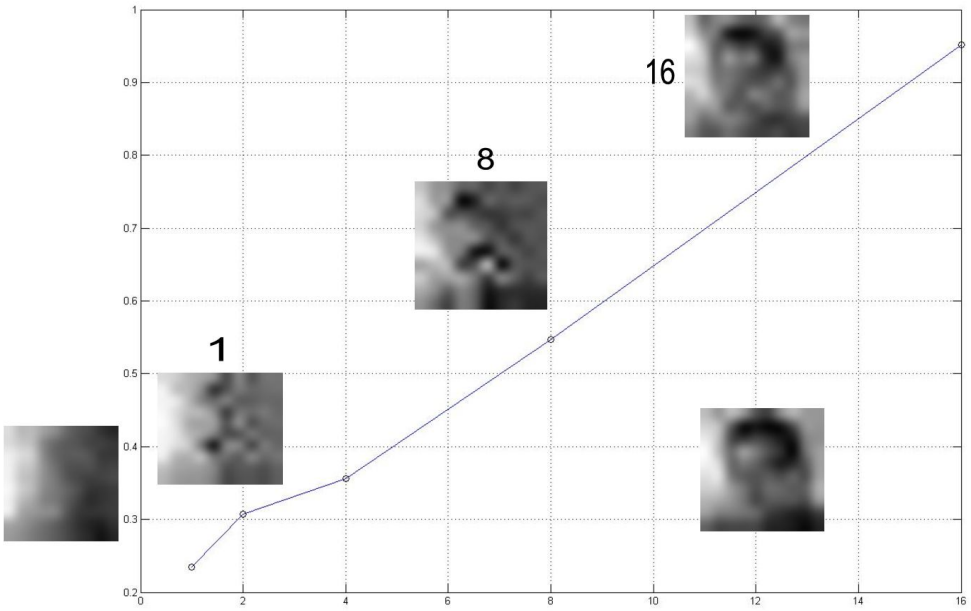


Figure 6.6: The plot show the average ratio of distances between the evolved and target patches to the distance between the initial and the target patch, computed on 15 different target patches with number of moments preserved on the x-axis. To the left of the y-axis is one of the target patches used, while the initial image is shown on the bottom right corner of the plot. Also shown along the plot are examples of evolved patches with number of constraints given on the top. The images are scaled up for visual purposes using Inkscape.

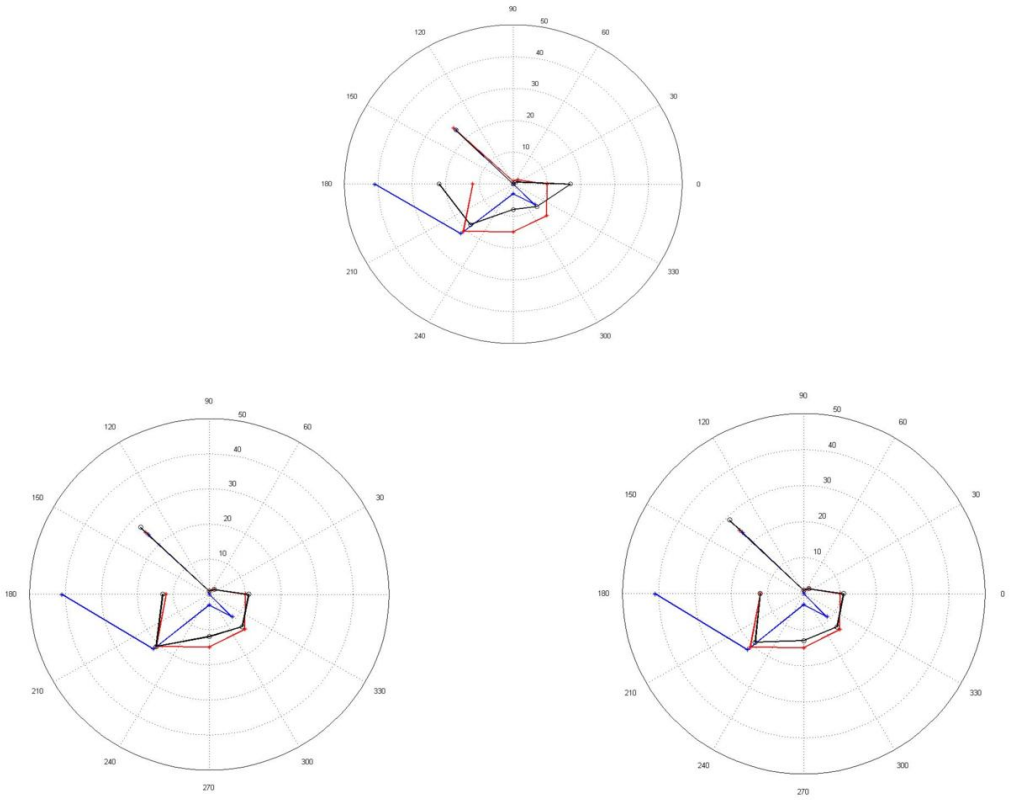


Figure 6.7: The histograms of the initial patch (plot in red with '*'), target patch (plot in blue with '*') and evolved patch (plot in black with 'o') for (top) 1,(bottom-left) 8 and (bottom-right) 16 number of moments preserved for the patch shown in the result before

with the same HoG feature. This shows that one has to be careful while working with the HoG features. We infer from the results that the metameric class of a patch may not be small.

This is a first step towards understanding the success of the SIFT feature. We still need to incorporate the idea of scale into our model. The feature points used in the SIFT feature are scale and space extremas of Laplacian of Gaussian of the image, while in our case we just assume that we are already given the feature point. We also know that our algorithm depends on the initial given patches u and w.

Chapter 7

Discussion and Conclusion

In this thesis we have given computational tools to restrict curve evolution to finite dimensional linear and implicitly defined nonlinear subspace of the space of curves. We can also do the same when the subspace is induced with a non-Euclidean inner product. We also develop the Log map for this case which can be used to compute geodesics and geodesic distance between two given curves in the nonlinear space. The Bicycle chain shape space can be useful for the purpose of medical image analysis. An approximation to the Exponential map is also built using gradient descent methods and is used for restricting curve evolution. We also use this method to study the equivalence class of patches with the same Histogram of Gradient Orientation features. We see that it is possible to evolve patches into a different patch preserving the features.

There are a number of questions still left to answer.

- The restricted curvature flow in both the linear and nonlinear subspace of Bicycle chain shapes, seems to evolve a simple planar curve into a round point, but we have not proved this.

- The space of Bicycle chain shapes seems to be interesting enough to demand a closer look at its structure. It is noteworthy that the Bicycle chain shape space with 4 node points becomes quite complicated due to various possible singular configurations (configurations where the rank reduces). Some representative shapes are given in Figure 7.1. It is interesting that a curve like example (a) in the figure has to pass (continuous deformation) through subspace of example (c) (dimension 1 subspace) in order to go to the subspace of configuration (b). There is a direct path from configuration (b) to configuration (c), (d) and (e). Again, there is no direct path from configuration (c) to configuration (d). A curve in subspace of configuration (d) can go to subspace of configuration (b) and (e), but not to that of (c).

- The Log map on Bicycle chain shape space using the non-Euclidean inner product using a gadient descent might not converge except in very local cases. We need to extend the algorithm to a more sound foundation. Also the Exponential map is computationally expensive at this point of time. An effecient implementation is needed to make it user friendly.

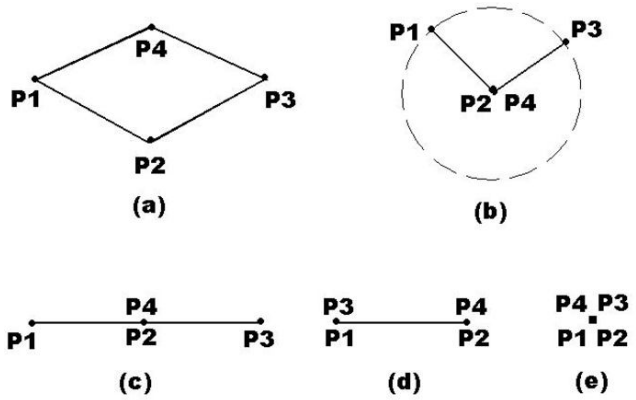


Figure 7.1: Some representative configurations in Θ^4. It is interesting to note that the dimension of the subspace of Θ^4 consisting of examples (a) and (b) is the same, and it reduces by 1 in case (c) and (d). The space collapses at the configuration (e)

- For the study of metameric class of Histogram of Gradient orientation based features, we need to look into other optimization techniques as well. Our initial aim was to get an intuition behind the success of the SIFT features. For this, we need to incorporate the scale of the image into the algorithm. We also need to consider the fact that SIFT features are computed at Laplacian scale and space extremas. This perhaps induces particular structure to the histogram. We also need to consider overlapping patches as in Dalal and Triggs [17].

Appendix A

Definitions from Differential geometry

We give definitions of some concepts from differential geometry that we use in the paper (mainly from [21]) for the convenience of the reader.

1. Differentiable Manifolds:
   A differentiable manifold of dimension n is a set M and a family of injective mappings $x_i : U_i \subset \mathbb{R}^n \to M$ of open sets U_i of $\mathbb{R}^n$ into M such that
   - $\bigcup_i x_i(U_i) = M$, i.e. the open sets cover M.
   - for any pair i, j with $x_i(U_i) \bigcap x_j(U_j) = W \neq \phi$, the mapping $x_j^{-1} \circ x_i$ is differentiable.
   - The family $\{U_i, x_i\}$ is maximal.

2. Immersion and Embedding:
   A differentiable mapping $\psi : M \to N$ between two manifolds is an immersion if its differential map $d\psi_p : T_pM \to T_{\psi(p)}N$ is injective for all $p \in M$. If ψ is also a homeomorphism from M onto $\psi(M) \subset N$ where $\psi(M)$ has the subspace topology of N, then ψ is called an embedding and $\psi(M)$ is an (embedded) submanifold of N.

3. Riemannian Metric:
   A Riemannian metric on a manifold M is a correspondence which associated to each point $p \in M$ an inner product $\langle -, - \rangle_p$ on the tangent space T_pM, which varies smoothly. In terms of local coordinates, the metric at each point x is given by a matrix, $g_{ij} = \langle X_i, X_j \rangle_x$, where X_i, X_j are tangent vectors to M at x, and it varies smoothly with x. A *Geodesic curve* is a local minimizer of arclength computed with a Riemannian metric.

4. Affine connection:
   Let $\mathscr{X}(M)$ be the set of all smooth vector fields on M. An affine connection ∇ on a differentiable manifold M is a mapping
   $$\nabla : \mathscr{X}(M) \times \mathscr{X}(M) \to \mathscr{X}(M)$$
   which is denoted by $\nabla(X, Y) \to \nabla_X Y$ and which satisfies the following properties:

- $\nabla_{fX+gY} Z = f\nabla_X Z + g\nabla_Y Z$.
- $\nabla_X (Y + Z) = \nabla_X Y + \nabla_X Z$.
- $\nabla_X (fY) = f\nabla_X Y + X(f)Y$.

in which $X, Y, Z \in \mathscr{X}(M)$ and f, g are $\mathcal{C}^\infty(M)$. This gives us a notion of directional derivative of a vector field defined on the manifold.

5. Covariant derivative:
   Let M be a differentiable manifold with affine connection ∇. There exists a unique correspondence which associates to a vector field V along the differentiable curve $c : I \to M$ another vector field $\frac{DV}{dt}$ along c, called the covariant derivative of V along c, such that
   - $\frac{D}{dt}(V + W) = \frac{DV}{dt} + \frac{DW}{dt}$, where W is a vector field along c.
   - $\frac{D}{dt}(fV) = \frac{df}{dt}V + f\frac{DV}{dt}$, where f is a differentiable function on I.
   - If V is induced by a vector field Y, a member of the tangent bundle of M, i.e. $V(t) = Y(c(t))$, then $\frac{DV}{dt} = \nabla_{\frac{dc}{dt}} Y$.

   In a parameterized manifold, where $c(t)$ is represented as $(x^1(t), \ldots, x^n(t))$, the covariant derivative becomes

$$\frac{Dv}{dt} = \sum_k \left\{ \frac{dv^k}{dt} + \sum_{i,j} \Gamma^k_{ij} v^i \frac{dx^i}{dt} \right\} \frac{\partial}{\partial x_k} \tag{A.1}$$

   where the Γ^k_{ij} are the *coefficients of the connection* also known as the *Christoffel symbols* Γ. For the Levi-Cività connection associated with the metric g of a Riemannian manifold, the corresponding Christoffel symbols are given by

$$\Gamma^k_{ij} = \frac{1}{2} \sum_m \left\{ \frac{\partial}{\partial x_i} g_{jm} + \frac{\partial}{\partial x_j} g_{mi} - \frac{\partial}{\partial x_m} g_{ij} \right\} g^{mk} \tag{A.2}$$

   g_{ij} is the ij^{th} element of the metric, and g^{ij} is the ij^{th} element of its inverse. A curve is geodesic if the covariant derivative of its tangent vector field is zero everywhere on it, which means that a geodesic curve has zero tangential acceleration. Such a curve c satisfies the second order system of ODEs, which, with the above parameterization becomes

$$\frac{d^2 x^k}{dt^2} + \sum_{ij} \Gamma^k_{ij} \frac{dx^i}{dt} \frac{dx^j}{dt} = 0, \quad k = 1 \ldots n. \tag{A.3}$$

6. Exponential map:
   The exponential map is a map $Exp : TM \to M$, that maps $v \in T_q M$ for $q \in M$, to a point on M obtained by going out the length equal to $|v|$, starting from q, along a geodesic which passes through q with velocity equal to $\frac{v}{|v|}$. Given $q \in M$ and $v \in T_q M$, and a parameterization $(x_1, \ldots, x_n)$ around q, $Exp_q(v)$ can be defined as the solution at time 1 of the above system of

ODEs (A.3) with initial conditions $(x^i(0)) = q$ and $(\frac{dx^i}{dt}(0)) = v$, $i = 1, \ldots, n$. The geodesic starting at q with initial velocity t can thus be parameterized as

$$t \mapsto Exp_q(tv).$$

7. Log map: For $\tilde{q}$ in a sufficiently small neighborhood of q, the length minimizing curve joining q and $\tilde{q}$ is unique as well. Given q and $\tilde{q}$, the direction in which to travel geodesically from q in order to reach $\tilde{q}$ is given by the result of the logarithm map $\mathrm{Log}_q(\tilde{q})$. We get the corresponding geodesics as the curve $t \mapsto \mathrm{Exp}_q(t\mathrm{Log}_q\tilde{q})$. In other words, Log is the inverse of Exp in the neighborhood.

References

[1] G. AUBERT AND P. KORNPROBST, *Mathematical Problems in Image Processing*, Springer, 2006.

[2] J. MALIK, S. BELONGIE AND J. PUZICHA, *Shape matching and object recognition using shape contexts*, vol. 24, 2002, p. 509522.

[3] A. BLAKE AND M. ISARD, *Active Contours*, Springer–Verlag, 1998.

[4] H. BLUM, *Biological shape and visual science*, Journal of Theoretical Biology, 38 (1973), pp. 205–287.

[5] F. BOOKSTEIN, *Morphometric Tools For Landmark Data: Geometry and Biology*, Cambridge University Press, 1991.

[6] W. M. BOOTHBY, *An Introduction to Differentiable Manifolds and Riemannian Geometry. Revised Second Edition*, Academic Press, 2003.

[7] P. BRIGGER, J. HOEG, AND M. UNSER, *B-Spline Snakes: A Flexible Tool for Parametric Contour Detection*, IEEE Transactions in Image Processing, 9 (2000).

[8] A. M. BRUCKSTEIN, G. SAPIRO, AND D. SHAKED, *Evolutions of planar polygons*, International Journal of Pattern Recognition and Artificial Intelligence, 9 (1995), pp. 991–1014.

[9] V. CASELLES, R. KIMMEL, AND G. SAPIRO, *Geodesic Active Contours*, International Journal of Computer Vision, 22 (1997), pp. 61–79.

[10] J. CATES, P. T FLECHTER, M. STYNER, M.E. SHENTON, AND R. WHITAKER, *Shape Modeling and Analyis with Entropy-Based Particle Systems*, in Proceedings of the 20th International Conference on Information Processing in Medical Imaging, Boudewijn P. F. Lelieveldt Nico Karssemeijer, ed., vol. 4584 of LNCS, Kerkrade, The Netherlands, 2007, Springer, pp. 333–345.

[11] G. CHARPIAT, O. FAUGERAS, AND R. KERIVEN, *Approximations of shape metrics and application to shape warping and empirical shape statistics*, Foundations of Computational Mathematics, 5 (2005), pp. 1–58.

[12] G. CHARPIAT, P. MAUREL, J.-P. PONS, R. KERIVEN, AND O. FAUGERAS, *Generalized gradients: Priors on minimization flows*, International Journal of Computer Vision, 73 (2007), pp. 325 – 344.

[13] C. CHEFD'HOTEL, D. TSCHUMPERLÉ, R. DERICHE, AND O. FAUGERAS, *Regularizing Flows for Constrained Matrix-Valued Images*, Journal of Mathematical Imaging and Vision, 20 (2004), pp. 73–87.

[14] R. CHELLAPPA AND R. BAGDAZIAN, *Optimal fourier coding of image boundaries*, IEEE Pattern Analysis and Machine Intelligence, 6 (1984), pp. 102 – 105.

[15] T. COOTES, C. TAYLOR, D. COOPER, AND J. GRAHAM, *Active shape models-their training and application*, Computer Vision and Image Understanding, 61 (1995), pp. 38–59.

[16] D. CREMERS, F. TISCHHAUSER, J. WEICKERT, AND C. SCHNÖRR, *Diffusion Snakes: Introducing Statistical Shape Knowledge Into the Mumford-Shah Functional*, International Journal of Computer Vision, 50 (2002), pp. 295–313.

[17] N. DALAL AND B. TRIGGS, *Histograms of oriented gradients for human detection*, 2005, pp. 886 – 893.

[18] R. H. DAVIES, C. J. TWINING, T. F. COOTES, J. C. WATERTON, AND C. J TAYLOR, *A Minimum Description Length Approach to Statistical Shape Modeling*, IEEE Transactions on Medical Imaging, 21 (2002), pp. 525–538.

[19] J.-P. DEDIEU AND D. NOWICKI, *Symplectic methods for the approximation of the exponential map and the newton iteration on riemannian submanifolds*, Journal of Complexity, 21 (2005), pp. 487 – 501.

[20] M.C. DELFOUR AND J.-P. ZOLÉSIO, *Shapes and Geometries*, Advances in Design and Control, Siam, 2001.

[21] M. P. DO CARMO, *Riemannian Geometry*, Mathematics: Theory and Applications, Birkhauser, 1992.

[22] H. EFSTATHIOS, M. D. GROSSBERG, AND S. K. NAYAR, *Histogram preserving image transformations*, International Journal of Computer Vision, 1 (2001), pp. 5 – 23.

[23] C. L. EPSTEIN AND M. GAGE, *The Curve Shortening Flow*, in Wave Motion:Theory, Modeling and Computation, Springer-Verlag, New York, 1987.

[24] P. T. FLETCHER, L. CONGLIN, S. M. PIZER, AND S. JOSHI, *Principal Geodesic Analysis for the Study of Nonlinear Statistics of Shape*, IEEE Transactions on Medical Imaging, 23 (2004), pp. 995 – 1005.

[25] P. J. GIBLIN AND B. B. KIMIA, *On the local forms and transition of symmetry sets, medial axis and shocks in 2d*, International Conference on Computer Vision, (1999), pp. 385–391.

[26] J. GLAUNÉS, A. QUI, AND M. I. MILLER, *Large deformation diffeomorphic metric curve mapping*, International Journal of Computer Vision, 80 (2008), pp. 317 – 336.

[27] J. GLAUNÈS, A. TROUVÉ, AND L. YOUNES, *Statistics and Analysis of Shapes*, Birkhäuser Boston, 2006, ch. Modeling planar shape variation via Hamiltonian flows of curves, pp. 335–361.

[28] C. GOODALL, *Procrustes methods in the statistical analysis of shape*, Journal of the Royal Statistical Society B, 53 (1991), pp. 285–339.

[29] L. GORELICK, M. GALUN, E. SHARON, R. BASRI, AND A. BRANDT, *Shape representation and classification using the poisson equation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 28 (2006), pp. 1991–2005.

[30] G.H. GRANLUND, *Fourier preprocessing for hand print character recognition*, IEEE Transactions on Computers, C-21 (1972), pp. 195–201.

[31] M. A. GRAYSON, *The Heat Equation Shrinks Embedded Plane Curves to Round Points*, Journal of Differential Geometry, 26 (1987).

[32] U. GRENANDER, Y. CHOW, AND D. M. KEENAN, *Hands: A Pattern Theoretic Study of Biological Shapes*, Springer Research Notes In Neural Computing, Springer-Verlag New York, Inc. New York, NY, USA, 1991.

[33] E. HAIRER, *Symmetric Projection Methods for Differential Equations on Manifolds*, BIT Numerical Mathematics, 40 (2000), pp. 726 – 734.

[34] ——, *Geometric integration of ordinary differential equations on manifolds*, BIT Numerical Mathematics, 41 (2001), pp. 996 – 1007.

[35] K. KAMARAJ AND K. C. SIVAKUMAR, *Moore-penrose inverse in an indefinite inner product space*, Journal of Applied Mathematics and Computing, 19 (2005), pp. 297 – 310.

[36] M. I. KAMIEN AND N. L. SCHWARTZ, *Dynamic Optimization: The Calculus of Variations and Optimal Control in Economics and Management*, Elsevier, 1991.

[37] H. KARCHER, *Riemannian center of mass and mollifier smoothing*, Communications on Pure and Applied Math., 30 (1977), pp. 509 – 541.

[38] M. KASS, A. WITKIN, AND D. TERZOPOULOS, *Snakes: Active Contour Models*, International Journal of Computer Vision, (1987), pp. 321–331.

[39] H. KAUPPINEN, T. SEPPNEN, AND M. PIETIKINEN, *An experimental comparison of autoregressive and fourier-based descriptors in 2-d shape classification*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 17 (1995), pp. 201–207.

[40] H. B. KELLER, *Numerical Methods for Two-Point Boundary-Value Problems*, Blaisdell Publishing Co., Waltham, Massachusets, USA, 1968.

[41] ——, *Numerical Solution of Two Point Boundary Problems*, in CBMF-NSF Regional Conference Series in Applied Mathematics, vol. 24, SIAM, 1976.

[42] D. G. KENDALL, *Shape manifolds, procrustean metrics, and complex projective spaces*, Bulletin of the London Mathematical Society, 16 (1984), pp. 81–121.

[43] D. G. KENDALL, D. BARDEN, T. K. CARNE, AND H. LE, *Shape and Shape Theory*, Wiley and Sons, 1999.

[44] R. KIMMEL, *Numerical Geometry of Images, Theory, Algorithms and Applications*, Springer-Verlag New York,Inc., Department of Computer Science, Technion-Israel Institute of Technology, Haifa 32000, Israel, 1st ed., 2004.

[45] E. KLASSEN AND A. SRIVASTAVA, *Geodesics Between 3D Closed Curves Using Path-Straightening*, vol. 3951 of Lecture Notes in Computer Science, Springer, 2006, pp. 95–106.

[46] E. KLASSEN, A. SRIVASTAVA, W. MIO, AND S. JOSHI, *Analysis of planar shapes using geodesic paths on shape spaces*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 26 (2004), pp. 372—383.

[47] J. M. LEE, *Riemannian Manifolds: An Introduction to Curvature*, Springer, 1997.

[48] D. G. LOWE, *Distinctive image features from scale - invariant keypoints*, International Journal of Computer Vision, 60 (2004), pp. 91 – 110.

[49] D. G. LUENBERGER, *Optimization by Vector Space Methods*, John Wiley & Sons, Inc., 1969.

[50] M. LILLHOLM, M. NIELSEN, AND L. D. GRIFFIN, *Feature-based image analysis*, International Journal of Computer Vision, 52 (2003), pp. 73 – 95.

[51] P. W. MICHOR AND D. MUMFORD, *Riemannian Geometries on Spaces of Plane Curves*, Journal of European Mathematical Society, 8 (2006), pp. 1– 48.

[52] P. W. MICHOR, D. MUMFORD, J. SHAH, AND L. YOUNES, *A metric on shape space with explicit geodesics*, Rend. Lincei Mat. Appl., 9 (2008), pp. 25–57.

[53] K. MIKOLAJCZYK AND C. SCHMID, *A performance evaluation of local descriptors*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 27 (2005), pp. 1615–1630.

[54] M. I. MILLER, A. TROUV, AND L. YOUNES, *Geodesic shooting for computational anatomy*, Journal of Mathematical Imaging and Vision, 24 (2006), pp. 209 – 228.

[55] M. I. MILLER AND L. YOUNES, *Group actions, homeomorphisms, and matching: A general framework*, International Journal of Computer Vision, 41 (2001), pp. 61 – 84.

[56] W. MIO AND A. SRIVASTAVA, *Elastic-string models for representation and analysis of planar shapes*, Proceedings of the IEEE Computer Society International Conference on Computer Vision and Pattern Recognition (CVPR), 2 (2004), pp. 10—15.

[57] L. NOAKES, *A global algorithm for geodesics*, Journal of the Australian Mathematical Society, 64 (1998), pp. 37–50.

[58] S. OSHER AND N. PARAGIOS, *Geometric Level Set Methods in Imaging, Vision and Graphics*, Springer-Verlag New York, 2003.

[59] A. PAPOULIS, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill Inc., 1991.

[60] X. PENNEC, *Probabilities and statistics on riemannian manifolds: Basic tools for geometric measurements*, Proc. of Nonlinear Signal and Image Processing, (1999), pp. 194 – 198.

[61] E. PERSOON AND K. S. FU, *Shape discrimination using fourier descriptors*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 8 (1986), pp. 388 – 397.

[62] L. S. PONTRYAGIN, V. G. BOLTYANSKII, R. V. GAMKRELIDZE, AND E. F. MISHCHENKO, *The Mathematical Theory of Optimal Processes*, Interscience Publishers, 1962.

[63] F. PRECIOSO AND M. BARLAUD, *B-spline active contour with handling of topology changes for fast video segmentation*, EURASIP Journal on Applied Signal Processing, 2002 (2002), pp. 555 – 560.

[64] A. PRESSLEY, *Elementary Differential Geometry*, Springer Undergraduate Mathematics Series, Springer-Verlag London Ltd., Department of Mathematics, King's College, The Strand, London WC2R 2LS, UK, first ed., 2002.

[65] J. ROSEN, *The gradient projection method for nonlinear programming, i, linear constraints*, SIAM J., 8 (1960), pp. 181–217.

[66] G. SAPIRO AND A. TANNENBAUM, *On area and length preserving geometric invariant curve evolutions*, IEEE Pattern Analysis and Machine Intelligence, 17 (1995), pp. 67 – 72.

[67] M. SARFRAZ, *Object recognition using fourier descriptors: Some experiments and observations*, International Conference on Computer Graphics, Imaging and Visualisation, (2006), pp. 281–286.

[68] F. R. SCHMIDT, M. CLAUSEN, AND D. CREMERS, *Shape matching by variational computation of geodesics on a manifold*, DAGM, (2006), pp. 142–151.

[69] J. SCHOENBERG, *Contributions to the problem of approximation of equidistant data by analytic functions*, Quart. Appl. Math, 4 (1946), pp. 45 – 99 and 112 – 141.

[70] T. B. SEBASTIAN, P. N. KLEIN, AND B. B. KIMIA, *Recognition of shape by editing shock graphs*, International Conference on Computer Vision, (2001), pp. 755–762.

[71] J. A. SETHIAN, *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Sciences*, Cambridge Monograph on Applied and Computational Mathematics, Cambridge University Press, 1999.

[72] J. A. SETHIAN, *Level Set Methods*, Cambridge Monograph on Applied and Computational Mathematics, Cambridge University Press, 1st ed., 1996.

[73] E. SHARON AND D. MUMFORD, *2d-shape analysis using conformal mapping*, International Journal on Computer Vision, 70 (2006), pp. 55 – 75.

[74] K. SIDDIQI, A. SHOKOUFANDEH, S. J. DICKINSON, AND S. W. ZUCKER, *Shock graphs and shape matching*, International Journal of Computer Vision, 35 (1999), pp. 13–32.

[75] J. E. SOLEM AND N. CHR. OVERGAARD, *A geometric formulation of gradient descent for variational problems with moving surfaces*, Scale-Space, (2005), pp. 419 – 430.

[76] S. SOMMER, A. TATU, C. CHEN, M. DE BRUIJNE, M. LOOG, D. JORGENSEN, M. NIELSEN, AND F. LAUZE, *Bicycle chain shape models*, CVPR Workshop on Mathematical Methods in Biomedical Image Analysis, (2009).

[77] A. SRIVASTAVA, W. MIO, E. KLASSEN, AND X. LIU, *Geometric analysis of constrained curves for image understanding*, Proc. 2nd IEEE International Workshop on Variational, Geometric and Level-Set Methods in Computer Vision (VLSM), (2003).

[78] G. SUNDARAMOORTHI, A. YEZZI, AND A. C. MENNUCCI, *Sobolev Active Contours*, International Journal of Computer Vision, 73 (2007), pp. 345 – 366.

[79] W. T. FREEMAN AND M. ROTH, *Orientation histograms for hand gesture recognition*, International Workshop on Automatic Face- and Gesture- Recognition, (1995), pp. 296 – 301.

[80] A. TATU, F. LAUZE, M. NIELSEN, AND O. FOGH OLSEN, *Curve Evolution in Subspaces*, in Scale Space and Variational Methods in Computer Vision, 2007.

[81] D. W. THOMPSON, *On Growth and Form*, 1992.

[82] M. UNSER, *Splines: A perfect fit for signal and image processing*, IEEE Signal Processing Magazine, (1999).

[83] M. UNSER, A. ALDROUBI, AND M. EDEN, *B-spline signal processing: Part 1 - theory*, IEEE Transactions on Signal Processing, 41 (1993), pp. 821–832.

[84] ——, *B-spline signal processing: Part 2 - efficient design and applications*, IEEE Transactions on Signal Processing, 41 (1993), pp. 834–848.

[85] X. WANG, L. HE, AND W. G. WEE, *Deformable contour method: A constrained optimization approach*, International Journal of Computer Vision, 59 (2004), pp. 87–108.

[86] Y. KE AND R. SUKTHANKAR, *Pca-sift: A more distinctive representation for local image descriptors*, 2004, p. 511 517.

[87] L. YOUNES, *Computable elastic distances between shapes*, SIAM Journal on Applied Mathematics, 58 (1998), pp. 565 – 586.

[88] C. T. ZAHN AND R. Z. ROSKIES, *Fourier descriptors for plane closed curves*, IEEE Trans. on Computers, 21 (1972), pp. 269 – 281.